Chapter 6
Integrating Knowledge Engineering with Knowledge Discovery in Database: TOM4D and TOM4L

Laura Pomponio and Marc Le Goc

Abstract Knowledge Engineering (KE) provides resources to build a conceptual model from experts' knowledge which is sometimes deficient to interpret the input data flow coming from a concrete process. On the other hand, data mining techniques in a process of Knowledge Discovery in Databases (KDD) can be used in order to obtain representative patterns of data which could allow to improve the model to be constructed. However, interpreting these patterns is difficult due to the gap which exists between the expert's conceptual universe and that of the process instrumentation. This chapter proposes then a global approach which combines KE with KDD in order to allow the construction of Knowledge Models for Knowledge Based Systems from expert knowledge and knowledge discovered in data. This approach is grounded in the Theory of Timed Observations on which both a KE methodology and a KDD process are based, so that the resulting models can be compared.

6.1 Introduction

A Knowledge Based System (KBS) carries out a set of knowledge intensive tasks for the purpose of putting in practice problem-solving capabilities, comparable to those of a domain expert, from an input data flow produced by a process.

In particular, a knowledge intensive task requires, by construction, a *Knowledge Model* in order to interpret the input data flow according to the task to be achieved, to identify an eventual problem to be solved and to produce a solution to this one.

L. Pomponio (✉) · M. Le Goc
Laboratoire de Sciences de l'Information et des Systèmes (LSIS), UMR CNRS 7296, Aix-Marseille University, Domaine universitaire de Saint Jérôme, Avenue Escadrille Normandie Niemen, 13397 Marseille, France
e-mail: laura.pomponio@lsis.org

M. Le Goc
e-mail: marc.legoc@lsis.org

C. Faucher and L. C. Jain (eds.), *Innovations in Intelligent Machines-4*,
Studies in Computational Intelligence 514, DOI: 10.1007/978-3-319-01866-9_6,

The Knowledge Engineering (KE) discipline provides methods, techniques and tools which facilitate and improve the modelling task of expert knowledge. In this field of study, most approaches model separately expert knowledge regarding the expert's reasoning mechanisms from expert knowledge specific to the domain of interest. Thus, a model of the expert's knowledge, called *Expert Model* (or *Knowledge Model*), obtained through this discipline will be generally made up of a model describing how the expert reasons about the process (a conceptual model of the expert's reasoning tasks) and of a representation of the knowledge used in the involved reasoning (a conceptual model of the domain knowledge). This latter is derived from the *Process Model* utilized by the expert in order to formulate his own knowledge. Knowledge Engineering allows then to establish a back and forth way between the expert's knowledge and the built *Expert Model* where the validity of this latter can be evaluated. However, two of the main drawbacks with the KE approaches are (1) the cost of knowledge acquisition and modelling process, which is too long for economic domains that use technologies with short life cycles and (2) the validation of the *Expert Model* which is mainly oriented to "case-based".

An interesting alternative to deal with these problems is to resort to the process of Knowledge Discovery in Database (KDD) which uses Data Mining techniques in order to obtain knowledge from data. In this approach, the process data flow is recorded by a program in a database where the data contained in such a database are analysed by means of Data Mining techniques in a KDD process with the purpose of discovering "patterns" of data. An n-ary relation among data can be considered a pattern when this relation has a power of representativeness according to the data contained in a database. This representativeness is related to a form of recurrence within the data; that is to say, an n-ary relation among data of a given set is a pattern, when this relation is ''often" observed in the database. Thereby, a set of patterns is then considered as the observable manifestation of the existence of an underlying model of the process data contained in the database. Nevertheless, establishing the meaning, regarding the expert's semantics, of such a *Data Model* entails a difficult task. One of the reasons for this difficulty is the deep difference between the universe of the process instrumentation, from where the data come, and the conceptual universe of the expert's reasoning where exist scientific theories and theirs underlying hypothesis. As a consequence, the validation of a *Data Model* is an intrinsically difficult task and a lot of work has to be done to constitute a knowledge corpus from a validated *Data Model.*

Thus, in this last decade the idea of combining Knowledge Engineering with Knowledge Discovery in Database emerges with purpose of taking the advantages of both disciplines in order to reduce the construction cost of suitable *Knowledge Models* for Knowledge Based Systems. The main idea is to make possible the cross-validation of an *Expert Model* and a *Data Model.* This aims to define a general perspective, by combining Knowledge Engineering with Knowledge Discovery in Database in a global approach of knowledge creation carried out from experts and knowledge discovered in data. The key point to achieve this is then to find a KE methodology and a KDD process which allow to produce *Expert*

Models and *Data Models* comparable each other by knowledge engineers and easily interpretable by experts.

As far as we know, only the KE methodology and the KDD process which are based on the Theory of Timed Observations [1] allow to compare their models each other. This theory has been established to provide a general mathematical framework for modelling dynamic processes from timed data by combining the Markov Chain Theory, the Poisson Process Theory, the Shannon's Communication Theory [2] and the Logical Theory of Diagnosis [3]. Thus, this theoretical framework provides the principles that allow to define a KE methodology, denominated TOM4D (Timed Observation Modelling For Diagnosis) [4–7], and a KDD process called TOM4L (Timed Observation Mining For Learning) [8–13]. Owing to that, both TOM4D and TOM4L are based on the same theory, the models constructed through both can be easily related and compared to each other.

The purpose of this chapter is to describe the way the Theory of Timed Observations builds a bridge between Knowledge Engineering and Knowledge Discovery in Database. In line with this aim, a global knowledge creation perspective which combines experts' knowledge with knowledge discovered in a database is presented. In order to show how models built through this perspective can be collated and complement each other, the proposed approach is applied to a very simple didactic example of the diagnosis of a vehicle taken from the book by Schreiber et al. [14].

The next section completes this introduction by presenting arguments about the need of a global approach which fuses Knowledge Engineering and Knowledge Discovery in Database. The main concepts of the Theory of Timed Observations are then introduced in order to present the TOM4D KE methodology and the basic principles of the TOM4L KDD process. Next, both TOM4D and TOM4L are applied to the didactic example above mentioned in order to show how the corresponding *Expert Models* and *Data Models* can be compared to each other. Finally, the conclusion section synthesizes this chapter and refers to some applications of our approach of knowledge creation on real world problems.

6.2 Two Knowledge Sources, Two Different Approaches

Creating or capturing knowledge can be originated from psychological and social processes or, alternatively, from data analysis and interpretation. That is to say, the two significant ways to capture knowledge are: synthesis of new knowledge through socialization with experts (a primarily people-driven approach) and discovery by finding interesting patterns through observation and intertwining of data (a primarily data-driven or technology-driven approach) [15].

6.2.1 Knowledge Engineering: A Primarily People-Driven Approach

Considering knowledge as intellectual capital in individuals or groups of them, the creation of new intellectual capital is carried out through combining and exchanging existing knowledge. With this perspective, Nonaka's *knowledge spiral* [16, 17], illustrated in Fig. 6.1, is considered in the literature as a foundational stone in knowledge creation. Nonaka characterizes knowledge creation as a spiralling process of interactions between explicit and tacit knowledge. The former can be articulated, codified, and communicated in symbolic form and/or natural language [18], the latter is highly personal and hard to formalize, making it difficult to communicate or share with others [19]. Each interaction between both existing knowledges gives as result new knowledge. Thus, this process is conceptualized in four phases: Socialization (the sharing of tacit knowledge between individuals), Externalization (the conversion of tacit into explicit knowledge: the articulation of tacit knowledge and its translation into comprehensible forms that can be understood by others), Combination (the conversion of explicit knowledge into new and more complex explicit knowledge) and Internalization (the

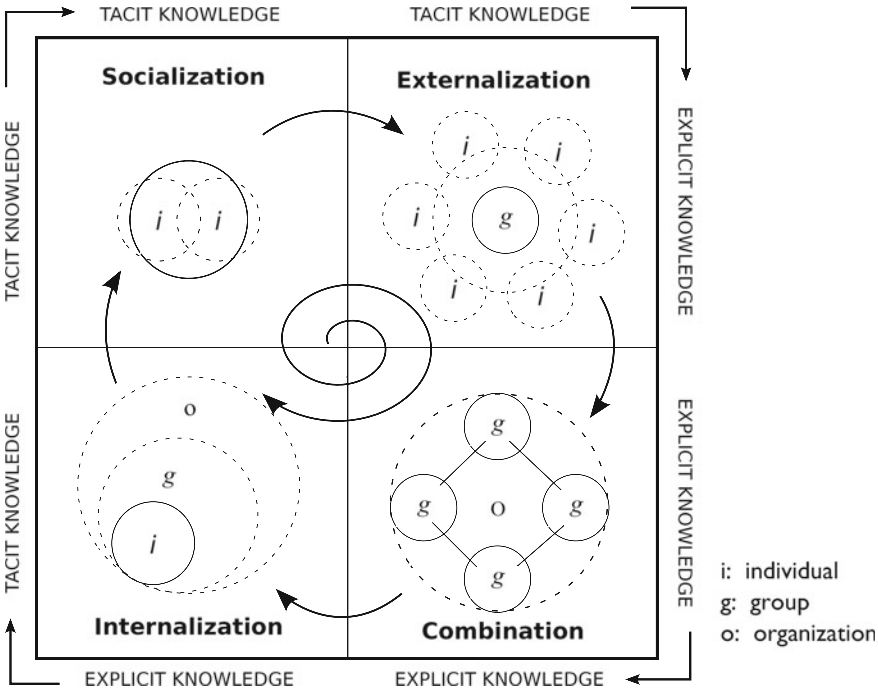


Fig. 6.1 Spiral evolution of knowledge conversion and self-transcending process [20, p. 43]

conversion of explicit knowledge into tacit knowledge: the individuals can broaden, extend and reframe their own tacit knowledge).

The tacit knowledge is, among other things, the knowledge of experts who intuitively know what to do in performing their duties but which is difficult to express because it refers to sub-symbolic skills. Such knowledge is frequently based on intuitive evaluations of sensory inputs of smell, taste, feel, sound or appearance. Eliciting such knowledge can be a major obstacle in attempts to build Knowledge Based Systems (KBSs). Knowledge Engineering (KE) arises then as the need of transforming the art of building KBSs into an engineering discipline [21, 22] providing thus techniques and tools that help to treat with the expert's tacit knowledge and to build KBSs. This discipline motived the development of a number of methodologies and frameworks such as Roles-Limiting Methods and Generic Tasks [23], and later, CommonKADS [14, 24], Protégé [25], MIKE [26, 27], KAMET II [28, 29] and VITAL [30]. In particular, CommonKADS is a KE methodology of great significance which proposes a structured approach in the construction of KBSs. Essentially, it consists in the creation of a collection of models that capture different aspects of the system to be developed, among which is the Knowledge Model (or Expert Model) that describes the knowledge and reasoning requirements of a system, that is, expert knowledge. Other two important modelling frameworks are MIKE and PROTÉGÉ, where the former focuses on executable specifications while the latter exploits the notion of ontology. All these frameworks or methodologies aim, of one or another way, to build a model of the expert's knowledge.

6.2.2 *Knowledge Discovery in Database: A Primarily Data-Driven Approach*

The traditional method of turning data into knowledge is based on data manual analysis and interpretation. For example, in the health-care industry, specialists periodically analyse trends and changes regarding health in the data. Then, they detail the analysis in a report which becomes the basis for future decision making in the domain of health. However, when data volumes grow exponentially and their manipulation is beyond human capacity, resorting to automatic analysis is absolutely necessary. Thus, computational techniques help to discover meaningful structures and patterns from data.

The field of Knowledge Discovery in Database (KDD) is concerned with the development of methods and techniques for making sense of data. The phrase *knowledge discovery in database* was coined at the first KDD workshop in 1989 [31] to emphasize that knowledge is the end product of a data-driven discovery. Although the terms KDD and Data Mining are often used interchangeably, KDD refers to the overall process of discovering useful knowledge from data, and Data Mining refers to a particular step in the mentioned process [32]. More precisely,

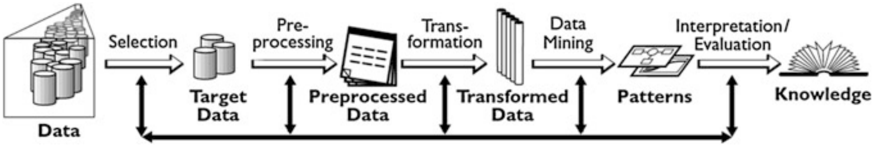


Fig. 6.2 Overview of the steps constituting the KDD process [33, p. 29]

this step consists of the application of specific algorithms in order to extract patterns from data.

The typical KDD process is depicted in Fig. 6.2 and summarized as follows [33]. The starting point is to learn the application domain and its goals. Next, to select a dataset or a subset of variables on which discovery is to be performed. Then preprocessing takes place which involves removing noise, collecting the necessary information to account for noise, deciding on strategies for handling missing data field, etc. The following step is data transformation which includes finding useful features to represent the data, depending on the goal of the task, and to reduce the effective number of variables under consideration or to find invariant representation for the data. After that, data mining is carried out. In general terms, this involves selecting data mining methods and choosing algorithms; and through these ones, searching for patterns of interest. Finally, the mined patterns are interpreted removing those that are redundant or irrelevant, and translating the useful ones into terms understandable by users. This discovered knowledge can be incorporated in systems or simply documented and reported to interested parties.

In a KDD process, finding patterns in data can be carried out through different techniques such as Decision Trees [34], Hidden Markov Chain [35], Neural Networks [36], Bayesian Networks [37], K Nearest-Neighbour [38], SVM [39], etc. All these techniques allow to obtain a model representative of the studied data where this model have to be interpreted and validated by expert knowledge.

6.2.3 The Need of One Integral Approach

The model-building of an observed process can be carried out through KE or KDD. As Fig. 6.3 depicts, 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. In turn, the process can be observed through sensors by a program which records data describing its evolution. 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. In an ideal world, both M_e and M_d would complement each other in order to have a process model M_{PR} more complete and suitable. That is, M_e must be validated with the process data perceived through sensors and M_d must be validated with expert knowledge. Nevertheless, some drawbacks arise. Knowledge Engineering approaches do not address the treatment of knowledge discovered in

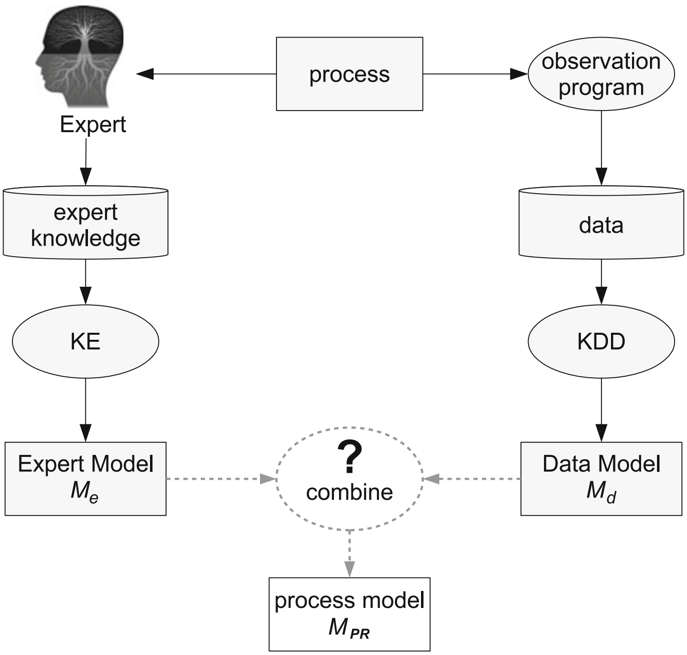


Fig. 6.3 Building a process model from two knowledge sources

databases, that is to say, sometimes the interpretation of discovered patterns is not trivial for an expert. Besides, relating models M_e and M_d obtained through KE and KDD, respectively, proves to be difficult owing to the different theories and the different natures of the representation formalisms used in both disciplines.

As [15] establishes, although capturing knowledge is the central focus of both fields of study, knowledge creation has tended to be approached from one or the other perspective, rather than from a combined perspective. Thus, a holistic view of knowledge creation that combines a people-dominated perspective with a data-driven approach is considered vital. In line with this need, this article proposes to integrate a KE methodology with data mining techniques in a KDD process in order to define a human–machine learning process.

6.3 Two Knowledge Sources, One Integral Approach

Models obtained through Knowledge Engineering (KE) and Knowledge Discovery in Database (KDD) will be able to be related and collated each other, if a bridge between the mentioned areas is established. We believe that fusing KE and KDD into a global approach of learning or knowledge acquisition, nourished with knowledge discovered in data and experts' knowledge, requires a theory on which to base both disciplines. The integral approach presented in this chapter and illustrated in Fig. 6.4 combines a KE methodology called Timed Observation Modelling For Diagnosis (TOM4D) [4–7] with a data mining technique, named Timed Observation Mining For Learning (TOM4L) [8–13]. Both TOM4D and TOM4L are based on the Theory of Timed Observations [1], a stochastic approach framework for discovering temporal knowledge from timed data.

The TOM4D methodology is a primarily syntax-driven approach for modelling dynamic processes where semantic content is introduced in a gradual and controlled way through the CommonKADS conceptual approach [14], Formal Logic and the Tetrahedron of States [40]. TOM4L is a probabilistic and temporal approach to discover temporal relations from initial timed data registered in a database. The time stamps are used to provide a partial order within the data in the database (i.e. two data can have the same time stamp) and to discover the temporal dimension of knowledge when needed. Owing to that, the underling theory is the same, TOM4D models and TOM4L models can be compared to each other in order to build a suitable model of the observed process. In particular, TOM4D allows to build a process model which, by construction, can be directly related to the knowledge model provided by the expert, i.e. a CommonKADS Knowledge Model; and besides, it can be collated with models obtained from data.

Figure 6.4 depicts the proposed overall view where a process model can be built through TOM4D from a knowledge source and then the constructed model can be validated by experts. In turn, an observation program $\Theta(X, \Delta)$ requires a model of the observed process for recording data in respect of the evolution of this one. These data are then analysed by means of TOM4L to produce a process model. This model can be directly related to the TOM4D model built from the expert's knowledge and consequently, it can be either validated by the expert or it

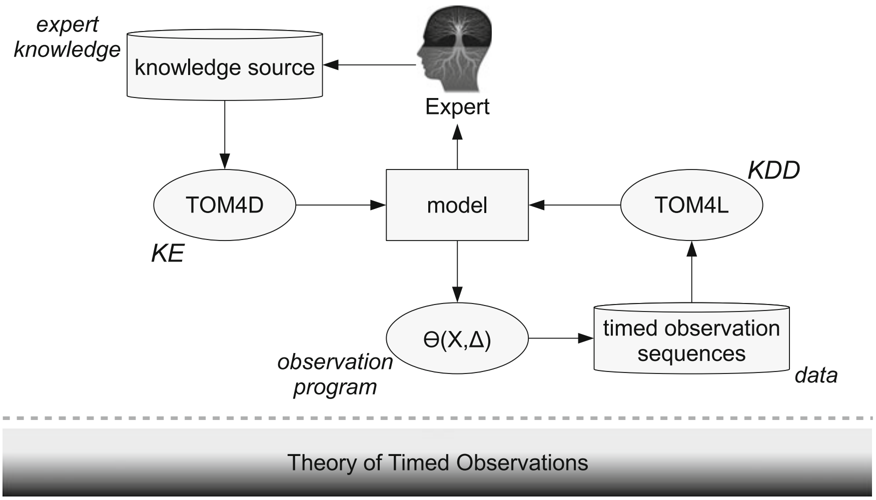


Fig. 6.4 Human-machine learning integral approach

can be utilized as pieces of new knowledge when the learning approach is applied to an unknown process. In this way, the built model can be defined through a back and forth way between experts' knowledge and knowledge discovered in data, establishing thus, an integral human–machine learning approach.

6.4 Introduction to the Theory of Timed Observations

The Theory of Timed Observations (TTO) [1] 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 [2] and the Logical Theory of Diagnosis [3]. The main concepts of the TTO, required in order to introduce the TOM4D KE methodology and the TOM4L KDD process, will be described in this section. These concepts are the notions of *timed observation* and *observation class*.

The Theory of Timed Observations defines a dynamic process as an arbitrarily constituted set $X(t) = \{x_1(t), \ldots, x_n(t)\}$ of n functions $x_i(t)$ of continuous time $t \in \Re$. The set $X(t)$ of functions implicitly defines a set $X = \{x_1, \ldots, x_n\}$ of n variable names x_i. The dynamic process $X(t)$ is monitored by a program $\Theta(X, \Delta)$ which observes the functions $x_i(t)$ of $X(t)$; and then, it establishes, records and informs their evolution over time with a finite set $\Delta = \left\{\delta_j\right\}_{j=1,\ldots,m}$ of constants δ_i (i.e. a number or a string). The program $\Theta(X, \Delta)$ usually accounts for the functions progression through messages recorded in a database. These messages can be alarms, warnings or reporting events.

This theory considers a message at time t_k as a *timed observation* (δ, t_k) where δ is a constant value of Δ and t_k is the moment in which the observation occurs. For example, let us suppose that timed data recorded in a database are of the form "yymmdd-hhmmss/message_value" where yymmdd-hhmmss is a time stamp and message_value is a value determined by a monitoring program. The message "080313-132225/TEMPERATURE/very_high" can be represented with a timed observation (δ, t_k) where t_k = 080313-132225 and δ = /TEMPERATURE/very_high. That is, $(\delta,\ t_k)$ = (TEMPERATURE/very_high, 080313-132225).

In general terms, a timed observation (δ, t_k) is written by an observer program $\Theta(\{x\}, \{\delta\})$ when a function $x(t)$ of continuous time enters in a specific interval of values. The specification of such an observer program refers to a threshold value $\Psi_j \in \Re$ and two immediately successive values $x(t_{k-1}) \in \Re$ and $x(t_k) \in \Re$ so that,

$$x(t_{k-1}) < \Psi_j \wedge x(t_k) \geq \Psi_j \Rightarrow write((\delta, t_k)). \tag{6.1}$$

In this program, *write(msg)* is a predicate which denotes that the element *msg* is recorded in a memory. For example, Fig. 6.5 illustrates a temperature function $x_i(t)$, where values above Ψ_j are interpreted by an observer program $\Theta(\{x_i\},$ {TEMPERATURE/very_high}) as very high temperature; that is, when $x_i(t) \in [\Psi_j, +\infty)$. Thus, given a sequence of values $w = (x_i(t_1), \ldots,\ x_i(t_{k-1}),$

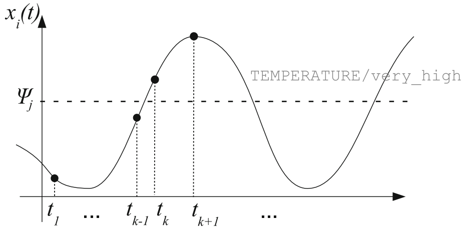


Fig. 6.5 Function of temperature

$x_i(t_k)$, $x_i(t_{k+1})$), the program $\Theta(\{x_i\}$,{TEMPERATURE/very_high}) will write a timed observation (TEMPERATURE/very_high, t_k), which indicates that the function $x_i(t)$ entered the interval $[\Psi_j, +\infty)$ at time t_k.

The Theory of Timed Observations establishes that the existence of a timed observation (δ, t_k), recorded in a database, allows to infer that the mentioned observation has been recorded by an unknown program $\Theta(\{x\}, \{\delta\})$ which implements the abstract logical equation described in (6.2).

$$\forall t_k \in \Gamma, \theta(x, \delta, t_k) \in \Theta \Rightarrow (\delta, t_k) \in \Omega \tag{6.2}$$

This sentence associates the set Θ of all the assignations 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 $\Theta(\{x\}, \{\delta\})$ (i.e., the database). A timed observation (δ, t_k) is then interpreted as the logical consequence of the assignation of the values x, δ and t_k to a ternary predicate $\theta(x_\theta,\ \delta_\theta,\ t_\theta)$. In other words, this means that the timed observation (δ, t_k) was recorded when the program $\Theta(\{x\}, \{\delta\})$ assigned the values x, δ and t_k to the predicate $\theta(x_\theta,\ \delta_\theta,\ t_\theta)$.

Given the sentences (6.1) and (6.2), the general meaning "**is**" can be always provided to the predicate θ so that the timed observation (δ, t_k) is interpreted as "**at time** t_k, x **is** δ". Considering that x is associated with a function $x(t)$, the meaning "**equal**" can also be attributed to the predicate θ, which leads to the following abuse of language: $\theta(x,\ \delta,\ t_k)$ means "$Equal(x,\ \delta,\ t_k)$" (i.e. "$x(t_k) = \delta$"). Consequently, the Theory of Timed Observations considers that a message contained in a database is a timed observation (δ, t_k) written by a program $\Theta(X, \Delta)$ which observes a time function $x(t)$ and implements the abstract Eq. (6.2). In our example, the timed observation (TEMPERATURE/very_high, t_k) indicates that a program $\Theta(x(t), \{\delta\})$, observing a time function $x_i(t)$ and defining implicitly a predicate $\theta(x_\theta,\ \delta_\theta,\ t_\theta)$, has considered $\theta(x_{i,}$ TEMPERATURE/very_high, t_k) true and then it has written the timed observation (TEMPERATURE/very_high, t_k) in the database Ω. This example illustrates the abuse of language frequently carried out, which associates the meaning "$x_i(t_k) =$ very_high" with the interpretation of the function "$x_i(t)$" as a temperature.

According to the Definition 6.1, the interpretation of a timed observation (δ, t_k) is precisely the assigned predicate $\theta(x,\ \delta,\ t_k)$. It is noteworthy that the program $\Theta(\{x\}, \{\delta\})$ could have errors; that is to say, a timed observation (δ, t_k) could have been written in a database although the assertion $\theta(x_i,\ \delta,\ t_k)$ is not really true.

Definition 6.1 Let $X(t) = \{x_i(t)\}_{i=1,\ldots,n}$ be a finite set of time functions; let $X = \{x_i\}_{i=1,\ldots,n}$ be the corresponding finite set of variable names; let $\Delta = \{\delta_j\}_{j=1,\ldots,m}$ be a finite set of constant values; let $\Theta(X, \Delta)$ be a program observing the evolution of the functions of $X(t)$; let $\Gamma = \{t_k\}_{k \in \Re}$ be a set of arbitrary time instants; and let $\theta(x_\theta, \delta_\theta, t_\theta)$ be a predicate implicitly determined by $\Theta(X, \Delta)$. Then,

- a *timed observation* $(\delta, t_k) \in \Delta \times \Gamma$ on $x_i(t)$ is the assignation of values x_i, δ and t_k to the predicate $\theta(x_\theta,\ \delta_\theta,\ t_\theta)$ such that $\theta(x_i,\ \delta,\ t_k)$;
- by definition $o(t_k)$ denotes a timed observation; i.e., $o(t_k) \triangleq (\delta, t_k)$[1] and,
- a finite set $O \subset \Delta \times \Gamma$ of timed observations is disjointly partitioned and ordered in a scenario Ω defined as a set of temporally ordered sequences of timed observations; that is, $\Omega = \{w : \{1,\ldots,n\} \to O\} | n \in \aleph \wedge \forall i,j \in \{1,\ldots,n\}, i<j, (w(i) = o(t_k) \wedge w(j) = o(t_r) \Rightarrow t_k \leq t_r)\} \wedge \bigcap_{w \in \Omega} \Im(w) = \emptyset \wedge \bigcup_{w \in \Omega} \Im(w) = O$ where $\Im(w)$ denotes the image or range of w, i.e. the observations of the sequence $w \in \Omega$.

Moreover, as follows from that previously explained, timed observations on a particular function implicitly determine a variable, which assumes discrete values and describes the function evolution according to an interpretation of the observer program. That is to say, when Θ considers $\theta(x_{i,}$ TEMPERATURE/very_high, $t_k)$ true and then writes (TEMPERATURE/very_high, t_k), it is implicitly defining a discrete variable which assumes the value TEMPERATURE/very_high. Consequently, a timed observation and the implicit existence of an associated discrete variable enable to define the notion of *observation class*, other important concept in this theory. An observation class associated with a variable x, that assumes values $\delta \in \Delta$, is a set $C_x = \{(x, \delta) \mid \delta \in \Delta\}$. For simplicity reasons, C_x is often defined as a singleton $C_x = \{(x, \delta)\}, \delta \in \Delta$. Thus, this concept establishes the link between a constant $\delta \in \Delta$ and a variable $x \in X$ and then, a timed observation (δ, t_k) is an occurrence of an observation class $C_x = \{(x,\ \delta)\}$. Definition 6.2 specifies this concept.

Definition 6.2 Let $X(t) = \{x_i(t)\}_{i=1,\ldots,n}$ be a set of time functions whose evolutions are observed by a program Θ; let $X = \{x_i\}_{i=1,\ldots,n}$ be a set of discrete variables where each x_i is associated with a time function $x_i(t)$ and its value is determined by an interpretation of Θ about the evolution of $x_i(t)$; and, let $\Delta = \bigcup_{x_i \in X} \Delta_{x_i}$ be such that Δ_{x_i} is a set of values which can be assumed by $x_i \in X$. Then we say that an *observation class* associated with a variable $x_i \in X$ is a set $C_i = \{(x_i, \delta) \mid \delta \in \Delta_{x_i}\}$.

In summary, from a message (TEMPERATURE/very_high, t_k) written in a database, the Theory of Timed Observations allows to consider that there exists a program $\Theta(\{x_i\}$,{TEMPERATURE/very_high}) which wrote the message, by means of observing a time function, maybe unknown for us, noted as $x_i(t)$. This

[1] The symbol $\triangleq$ denotes rewriting or "corresponds to".

message is then a timed observation (TEMPERATURE/very_high, t_k) indicating that a certain predicate $\theta(x_{i,}$ TEMPERATURE/very_high, t_k) was assumed true by the program $\Theta(\{x_i\}$,{TEMPERATURE/very_high}). Then, there is tacitly a discrete variable x_i which takes at least the value TEMPERATURE/very_high. Therefore, we can define an observation class $C_i = \{(x_i,$ TEMPERATURE/very_high)}, so that the timed observation (TEMPERATURE/very_high, t_k) is an occurrence of C_i. When knowing that the time function $x_i(t)$ represents the evolution of temperature, it is inferred that (1) x_i denotes a *variable of temperature*, (2) the observation class C_i can then be written as $C_i = \{(\text{very\_high})\}$ denoting that the *temperature is very high* and (3) the timed observation (TEMPERATURE/very_high, t_k) is an occurrence of this class, which means "at time t_k, temperature is very high".

For 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 [41]. Such a model can be as simple as the switch of an interrupter or requiring complex techniques, as signal processing techniques for artificial vision.

The TOM4D KE methodology and the TOM4L KDD process are based on these notions of timed observation and observation class, as the next sections describe below.

6.5 TOM4D KE Methodology

TOM4D is a modelling approach for dynamic systems focused on timed observations. The objective of this one is to produce suitable models for dynamic process diagnosis from timed observations and experts' a priori knowledge. This methodology combines then the modelling of the experts' cognitive process, using CommonKADS [14, 24], with a multi-modelling approach for dynamic systems [40, 42]. In addition, TOM4D is a primarily syntax-driven approach [5–7] which resorts to CommonKADS, Formal Logic and the Tetrahedron of States (ToS) [40] as interpretation frameworks and paradigms in order to introduce, in the modelling process, semantic content in a gradual and controlled way.

6.5.1 Multi-Modelling

In this methodology, a system is represented by means of four models, the three models described in the conceptual multi-modelling framework introduced in [43] and a complementary model called *Perception Model* [6].

The models of the multi-modelling framework are *Structural Model* (*SM*), *Behavioural Model* (*BM*) and *Functional Model* (*FM*) which describe different types of knowledge. The *SM* contains knowledge relative to the system components and their structural organization, that is to say, the relations between these ones. The *BM* specifies knowledge about the phenomena which act inside the system in order to transform an input flow into an output flow. Such transformations are measured through the evolution of the values of a set of variables. Thus, these changes in the values define the possible sequences of observation classes that can occur and therefore, the discernible states between them. Finally, the *FM* describes knowledge about the relations among the values that the variables can assume.

For its part, the *Perception Model* (*PM*) contains knowledge about the following elements and aspects of the process: variables and their thresholds, operating goals, and normal and abnormal operating modes.

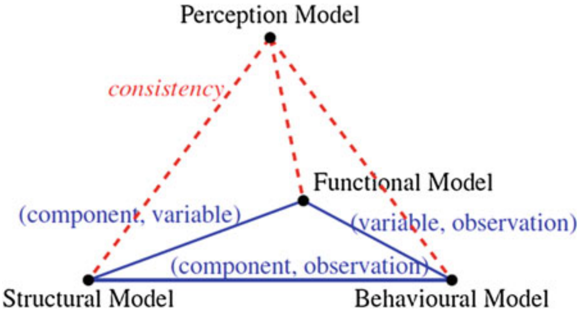
The relations between the first three models are determined by the notion of *variable* as Fig. 6.6 illustrates. A variable used in a function of the *Functional Model* is associated with a component of the *Structural Model* and, a discrete event of the *Behavioural Model* is the assignment of a value to the variable. Indeed, any specification in these models must be consistent with that one made in the *Perception Model*.

6.5.2 Interpretation Frameworks

CommonKADS [14, 24] is a methodology which offers a structured approach in the development of KBSs by proposing three groups of models. The first group regarding the organizational context and environment, the second one with respect to the conceptual description of the knowledge applied in a task, and the last one concerning the technical aspects of the software artefact.

In particular, the CommonKADS Knowledge Model which belongs to the second group is utilized 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. This model is developed, in a way that is understandable by humans, as part of the analysis process and therefore, it does not contain any

Fig. 6.6 Relations between TOM4D models



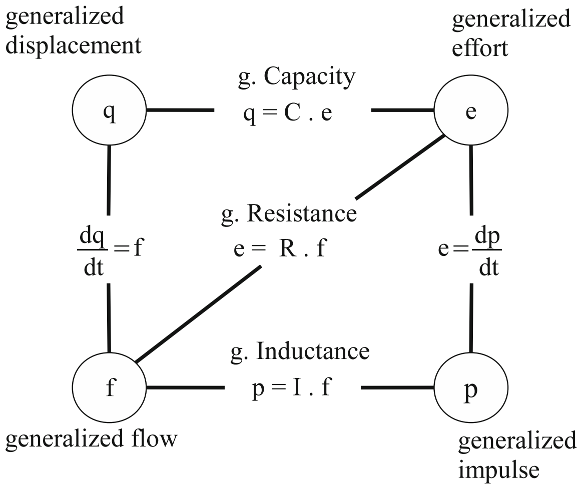


Fig. 6.7 Tetrahedron of states (ToS) (based on [40, p. 1728])

implementation-specific term. Thus, this one is an important vehicle for communication with experts and users about the problem-solving aspects. Consequently, TOM4D uses the aforementioned model as a mean of interpreting and structuring the available knowledge.

Formal logic is also used by the proposed methodology as a resource which provides reasoning mechanisms and gives the possibility of utilizing Reiter's Theory of Diagnosis [44]. In turn, in order to give a physical interpretation to the variables, the Tetrahedron of States (ToS) [40, 45, 46] can be incorporated in the analysis process. The ToS is a framework that describes a set of generalized equations (Fig. 6.7) which are common to a wide variety of physical domains (electromagnetism, fluid dynamics, thermodynamics, etc.). This one allows to map physical variables of a specific domain into four classes of generalized variables (*effort*, *flow*, *impulse* and *displacement*) and to identify the set of relationships among these ones. For example, in the electric domain (Electric ToS), *current* is mapped to generalized *flow*, *electric charge* to generalized *displacement*, *voltage* to generalized *effort* and *magnetic flux* to generalized *impulse*; thus, the relations among the electric domain variables can be established according to the ToS. Our modelling approach then resorts to Formal Logic and ToS as paradigms of interpretation and analysis of knowledge.

6.5.3 TOM4D Modelling Process

The modelling approach of this methodology is based on three principles [7]. The first one is that each symbol of an entity used in one of the three models introduced in Sect. 6.5.1 (structural, functional and behavioural models) denotes a concept that is defined at the level of domain knowledge of a CommonKADS model [14].

This means that the introduction of a symbol that is not associated with an element of the domain knowledge model is prohibited. The second principle is that a variable is always associated with a component or a component aggregate defined in the structural model. The third principle is that a transition between two states is conditioned by the assignment of a new value to a variable. The notion of variable, as aforementioned in Sect. 6.5.1, constitutes thus the common point of the three models.

The modelling process aims to produce a generic model of a system from available knowledge and data, where the three fundamental modelling phases are **knowledge interpretation**, **process definition** and **generic modelling**. Figure 6.8 illustrates a structure of logical dependences that describe the TOM4D reasoning process for obtaining a model of an observed system. Therefore, how the control flow of the modelling process is carried out, is not part of this structure. The illustrated process, introduced below, gives a general guide in order to understand the principal objectives of this approach. Clearly, the modelling is generally

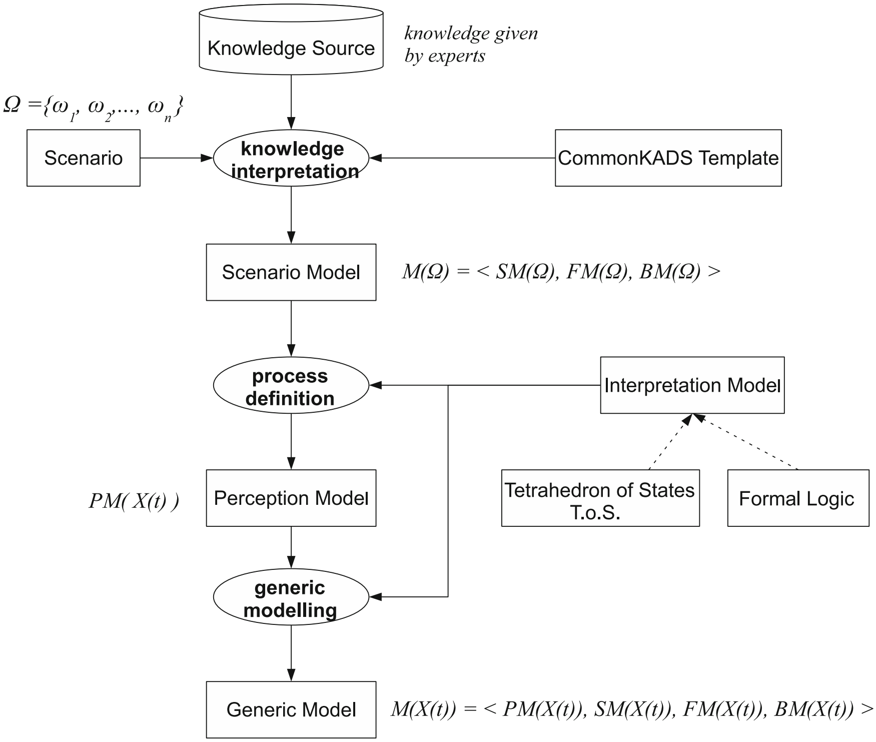


Fig. 6.8 General structure of the TOM4D modelling process

cyclical and each stage can require to return to previous phases with the objective of revising the expert's knowledge, results, ideas, modelling decisions, etc.

1. **Knowledge Interpretation**

The objective of this phase is to define a scenario model. In general terms, a scenario Ω of a system is a set of observations or measures over time on the variables of the system, where these measures describe a certain evolution of the process that drives the system dynamic. Definition 6.1 in Sect. 6.4 introduces the meaning of *scenario* and other concepts such as *timed observation* and *observation class*. In short, a scenario is a set of sequences of timed observations describing partially the behaviour of a process.

The construction of a scenario model $M(\Omega) = < SM(\Omega), FM(\Omega), BM(\Omega) >$ consists of the definition of a structural model $SM(\Omega)$, a functional model $FM(\Omega)$ and a behavioural one $BM(\Omega)$ of Ω.

For the purpose of defining a model $M(\Omega)$, a CommonKADS template is utilized to interpret and to organize the available knowledge. This knowledge is provided by a scenario Ω and a knowledge source where the latter can be an expert, a set of documents, etc. Thus, the outcome of this phase is an organized description of knowledge and available information.

2. **Process Definition**

The process definition step aims to define the process $X(t)$ that governs a system; that is, the boundary of the process, the operating goals and the normal and abnormal operating modes of this one. In this phase, the available knowledge, the scenario model $M(\Omega)$ and the concepts of Formal Logic or the Tetrahedron of States (ToS) can be used to achieve the objective. As described in Sect. 6.5.2, the last two are interpretation frameworks which allows, along with CommonKADS, to introduce semantic content in a controlled way, providing contexts of logical and physical interpretation of variables. The result of this phase must then be a perception model of the process, that is, $PM(X(t))$.

3. **Generic Modelling**

This stage aims to define a generic model of a process $X(t)$. The definition of this model consists of the perception model defined in previous steps and structural, functional and behavioural models associated with the process $X(t)$; that is, $M(X(t)) = < PM(X(t)), SM(X(t)), FM(X(t)), BM(X(t)) >$. The objective is then to define a model already not relative to a particular scenario Ω, but to a type of process. This model should be more general and more abstract than the scenario model and thus, more useful for diagnosis. This stage can be accomplished using the available knowledge, the *Perception Model* and analyses through Formal Logic and the ToS.

The results of applying TOM4D to a didactic example will be presented later in order to show how the built TOM4D model can be related to a TOM4L model automatically obtained from data.

6.6 TOM4L KDD Process

TOM4L [12], based on the Theory of Timed Observations [1], is a probabilistic and temporal approach to discover temporal relations for description, diagnosis and prediction from initial timed data Ω registered in a database (i.e. a set of timed observation sequences). The aim is to discover n-ary temporal relations which are representative of the process behaviour which gave rise to Ω.

In particular, the TOM4L approach is implemented by the ElpLab Java software, so that the n-ary temporal relations can be discovered in an automatic way.

6.6.1 Temporal Relations

As described in Sect. 6.4, sequences of timed observations $(\delta, t_k) \in \Delta \times \Gamma$ recorded by a program observing a process allow to establish a set of discrete variables $x \in X$; and consequently, a set C of corresponding observation classes $C_i \in C$. For example, if $C_{1a} = \{(x_i, \delta_a)\}$ is defined as an observation class associated with x_i, then a timed observation (δ_a, t_k) is an occurrence at time t_k of the class C_{1a}. In order to specify that an observation is of a certain class, the symbol '::' is used; e.g., $(\delta_a, t_k) :: C_{1a}$.

TOM4L aims to discover temporal characteristics present in the data that describe the evolution of a process; therefore, detailed descriptions about variables and particular values that these variables can assume are not necessary in this context. In particular, we shall refer to timed observations and observation classes. We recall that the timed observation (δ_a, t_k) can be rewritten as $o(t_k)$ (Definition 6.1); thus, we refer to this observation like $o(t_k)$ and we specify its class with the symbol '::' like $o(t_k) :: C_{1a}$.

A temporal relation between two observation classes describes a temporal constraint between observations of the involved classes. By considering $I = \{[\tau^-, \tau^+] \mid [\tau^-, \tau^+] \subset \Re\}$ a set of time intervals and C a set of observation classes, a temporal relation between two observation classes is a pair $(q, \bar{i})$ where $q \in C \times C$ and $\bar{i} \in I$. Thus, a temporal relation $(q, \bar{i}) = ((C_i, C_j),\ [\tau^-, \tau^+])$ specifies a temporal constraint between timed observations of the observation classes $C_i, C_j \in C$. Figure 6.9 illustrates this relation according to the ElpLab representation.

In particular, two observations verify the aforesaid relation if 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^+. That is to say, two observations $o(t_k), o(t_r) \in \Delta \times \Gamma$ verify the relation $((C_i, C_j),\ [\tau^-, \tau^+])$ if $o(t_k) :: C_i \wedge o(t_r) :: C_j \wedge (t_r - t_k) \in [\tau^-, \tau^+]$.

For its part, an n-ary temporal relation is a sequence m of temporal relations. Thus, a sequence of timed observations verifies an n-ary temporal relation m if the mentioned sequence verifies each temporal relation in m, even if in the middle of

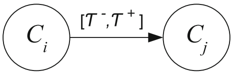


Fig. 6.9 Binary temporal relation $((C_i, C_j), \; [\tau^-, \tau^+])$ between two observation classes

the observation sequence there exist occurrences of classes that are not present in m.

As an example, we consider the observation classes C_{1a}, C_{2b}, C_{3c} and the n-ary temporal relation $m = (((C_{1a}, C_{2b}), [2, 5]), \; ((C_{2b}, C_{3c}), [0, 4]))$, as illustrated in Fig. 6.10. Besides, we suppose the sequence of timed observations $w = (o(19), o(20), o(22), o(24))$ such that $o(19) :: C_{1a}$, $o(20) :: C_{3c}$, $o(22) :: C_{2b}$ and $o(24) :: C_{3c}$, also illustrated in the figure. In this case, w verifies m owing to the following. Firstly, the class of the first observation coincides with the first class in the n-ary relation (i.e., $o(19) :: C_{1a}$) and the class of the last observation in w coincides with the last class in m (i.e., $o(24) :: C_{3c}$). In addition, the sequence of relations $m = (((C_{1a}, C_{2b}), [2, 5]), \; ((C_{2b}, C_{3c}), [0, 4]))$ is verified in w. That is to say, $((C_{1a}, C_{2b}), [2, 5])$ specifies that the elapsed time between an occurrence of the observation class C_{1a} and an observation of the class C_{2b} is greater than or equal to 2 and less than or equal to 5. Thus, in w, $o(19)$ and $o(22)$ verify this temporal constraint since that $o(19) :: C_{1a}$, $o(22) :: C_{2b}$, $22 - 19 = 3$ and $2 \leq 3 \leq 5$. In a similar way, $o(22)$ and $o(24)$ verify $((C_{2b}, C_{3c}), [0, 4])$. It is noteworthy that between $o(19)$ and $o(22)$, the observation $o(20)$ takes place. However, this does not invalidate that the relation $((C_{1a}, C_{2b}), [2, 5])$ is verified, along with the complete n-ary relation, in the sequence of observations w.

In this way, given a set of data describing the behaviour of a process, discovering the n-ary temporal relations that are representative of these data is the central focus in the TOM4L KDD process.

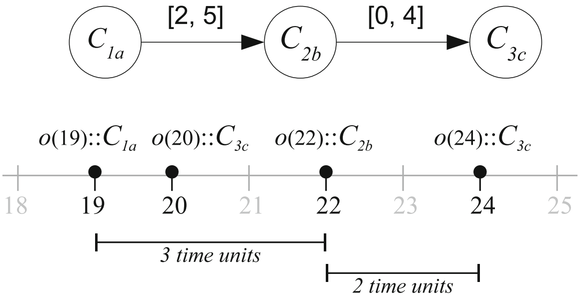


Fig. 6.10 Sequence $w = (o(19), o(20), o(22), o(24))$ of timed observations that satisfies the n-ary temporal relation $m = (((C_{1a}, C_{2b}), [2, 5]), \; ((C_{2b}, C_{3c}), [0, 4]))$

6.6.2 Stochastic Approach

In TOM4L, the analysis of a sequence w of timed observations consists of finding the more representative sequential relations between observation classes and establishing the temporal constraints in each relation. Thus, the study of the mentioned relations is addressed by resorting the Markov chain theory and the estimation of temporal constraints is dealt with the Poisson process theory. Consequently, in this framework, a sequence w of timed observations has a *stochastic representation* that consists of associating with w a superposition of the Poisson process and a Markov chain.

Given a finite set $O \subset \Delta \times \Gamma$ of timed observations, w is the sequence of all observations in O (i.e., the image of w is equal to O, or $w : \aleph \rightarrow O$ and $\Im(w) = O$) and C is the set of the n classes of observations in w. A stochastic representation of w consists then of a set of matrices reflecting different properties, where the rows and columns refer to the observations classes in C; that is to say, matrices $n \times n$ where the element of row i, column j refers to the sequential relation between the class C_i and the class C_j. We denote by $P(C_j \mid C_i)$ the conditional probability $P(w(k) :: C_j \mid w(k-1) :: C_i)$ of observing an occurrence of C_j having immediately before observed an occurrence of C_i and we denote by $P((C_i\ ,\ C_j))$ the probability $P((w(k-1) :: C_i\ ,\ w(k) :: C_j))$ of observing an occurrence of C_i followed immediately by an occurrence of C_j. Thus, the stochastic representation of w is given by the set of the following matrices. $\mathrm{N} = (N_{ij})_{n \times n}$ is a matrix where each N_{ij} establishes the number of observations of C_i followed immediately by an observation of C_j in w. The matrix $\mathrm{P} = (p_{ij})_{n \times n}$ establishes the transition probabilities between two observation classes, where the value p_{ij} corresponds to $P(C_j \mid C_i)$ and is calculated, based on N, as the rate between the number of the occurrences of C_i followed immediately by an occurrence of C_j and the number of occurrences of C_i followed immediately by an occurrence of any class.

The temporal constraints between two observation classes are calculated by analysing only the two subsequences of w whose observations are of the classes in question. In other words, w is partitioned in a set Ω of sequences w_r, where the observations in each w_r are of a same class C_r. By considering $w_i, w_j \in \Omega$ the subsequences of w whose observations are of the classes C_i and C_j respectively, the temporal constraint $[\tau^-, \tau^+]$ of a relation $((C_i, C_j),\ \ [\tau^-, \tau^+])$ is computed from the average of the elapsed times between an observation of class C_i and the following and first observation of class C_j, when overlapping w_i and w_j.

Based on theses calculations, an algorithm called BJT computes the stochastic representation of a sequence w under study, and an algorithm called BJT4T, based on the mentioned representation and on an abductive reasoning, builds a three of n-ary temporal relations associated with a given observation class C_i, i.e., paths ended in C_i representative of w. Both algorithms belonging to the TOM4L framework are implemented by ELpLab.

6.6.3 BJ-Measure and the Bayesian Networks Building

The BJ-measure [12] is a measure based on information entropy. Considering a superimposition of occurrences of two timed observation classes, this measure allows to evaluate the strength of intertwining of the mentioned superimposition; that is to say, the strength of an oriented binary relation between two observation classes taken from an arbitrarily built set.

Given an ordered binary relation $(C_i,\ C_j)$ between two observation classes, if these classes are independent, the probability of observing an occurrence of C_j at a time t_k is equal to the probability of observing an occurrence of C_j at that time having observed an occurrence of class C_i at the previous time t_{k-1}; that is, $P(C_j \mid C_i) = P(C_j)$. However, according to [12], if the classes are not independent, an occurrence of C_i at a time t_k provides information about an occurrence (or not) of C_j at the subsequent time t_{k+1}. In particular, the interest is in a measure indicating that an occurrence of class C_i at the time t_k increases the probability of observing an occurrence of class C_j at the time t_{k+1}; that is, $P(C_j \mid C_i) \geq P(C_j)$. Thus, the BJ-measure is based on the Kullback–Leibler distance [47] between two probability distributions which can be interpreted as the amount of information lost when a probability distribution is approximated by another distribution. The general idea is then the analysis, on the base of this distance measure, of the distance between $P(C_j \mid C_i)$ and $P(C_j)$ in order to establish if the relation $(C_i,\ C_j)$ is strong or weak.

Consequently, the BJ-measure $BJM(C_i,\ C_j)$ is defined from associating a sequential relation $(C_i,\ C_j)$ with a discrete memoryless communication channel [2] and from using the Kullback–Leibler distance. In particular, the BJ-measure verifies the properties of monotony, dissymmetry, positivity, independence and triangular inequality. Thus, if $BJM(C_i,\ C_j)$ is negative, the relation $(C_i,\ C_j)$ is weak; otherwise it is considered a possibly strong relation, or of interest.

The maximum and minimum values of the BJ-Measure depend on the rate $\tilde{\theta}_{ij}$ between the number of observations of class C_i and the number of observations of class C_j in the studied sequence w. In particular, [12] shows that a sequential relation $(C_i,\ C_j)$ is credible in the sense of the BJ-Measure if and only if $\frac{1}{4} < \tilde{\theta}_{ij} < 4$. This condition allows to select then relations of interest that provide a representative model of the sequence w.

TOM4L proposes an algorithm called Tom4BN [10, 11] to build Naive Bayesian Networks from timed data. Inspired by Cheng et al.'s algorithm [48], Tom4BN uses the properties of monotony, dissymmetry, positivity, independence and triangular inequality of the BJ-Measure to build a Naive Bayesian Network.

The general idea of the Tom4BN algorithm is to remove the sequential relations $(C_i,\ C_j)$ that are not of interest when building, from a set of timed data, the structure of a Naive Bayesian Network associated with a given observation class. For example, if $R_{BN} \subseteq C \times C$ is the set of all binary sequential relations $(C_i,\ C_j)$ with which paths in a Bayesian Network can be built, in principle $R_{BN} = C \times C$;

then, relations (C_i, C_j) where $BJM(C_i, C_j) \leq 0$ or $BJM(C_i, C_j) < BJM(C_j, C_i)$ are removed from R_{BN}. These and other criteria based on the aforesaid properties allow to select the binary sequential relations suitable for building a Bayesian Network from a data set. Consequently, given a goal class, the structure of the Bayesian Network associated with this one is constructed by the aforementioned algorithm from the mentioned criteria.

Based on properties which follow from the discrete memoryless communication channel, the tables of conditional probability for the Bayesian Network are defined from the matrix $\mathrm{N} = (N_{ij})_{n \times n}$ established by the stochastic representation. That is to say, are defined the probability $P(C_i)$ of a root node, the probability $P(C_i, C_j)$ for a simple sequential relation and the probabilities for two sequential relations (C_i, C_j) and (C_z, C_j) associated with the same class C_j (i.e., $P(C_j \mid C_i, C_z)$, $P(C_j \mid \neg C_i, C_z)$, $P(C_j \mid C_i, \neg C_z)$, $P(C_j \mid \neg C_i, \neg C_z)$). Thus, from the mentioned definitions, probabilities as for example $P(\neg C_j \mid C_i)$ can be calculated as $P(\neg C_j \mid C_i) = 1 - P(C_j \mid C_i)$; and $P(\neg C_j \mid C_i, \neg C_z)$ as $P(\neg C_j \mid C_i, \neg C_z) = 1 - P(C_j \mid C_i, \neg C_z)$.

Thereby, given a goal class, the Bayesian Network associated with this one can be automatically built through the Tom4BN algorithm from a data set.

6.6.4 Signatures

An n-ary temporal relation *m* is considered representative of a sequence *w* of timed observations from evaluating two rates, *anticipation rate* and *coverage rate* [13].

Considering *w* a sequence of observations, let *m* be a sequence of temporal relations and let m_s be the sequence resultant of eliminating from *m* the last binary relation. The anticipation rate T_A of *m* in *w* is the rate between the number of subsequences w_j of *w* that satisfy *m* (i.e., $w_j \sqsubseteq w \wedge satisfies(w_j, m)$) and the number of subsequences of *w* that satisfy m_s, as illustrated in Fig. 6.11. That is to say, the percentage of cases in which after observing an instance of m_s, an occurrence of the last relation in *m* takes place. Clearly, T_A is of great interest in the diagnosis task when allowing to anticipate the occurrence of an observation class, in particular the last class of the model *m*; i.e., C_i in Fig. 6.11.

In the TOM4L framework, a *signature* [13] is a model *m* that has certain representativeness in the data, that is, in the sequence *w* under study. In particular, this representativeness is given when the anticipation rate T_A is above a certain value T_{Amin} (typically, 50 %). In other words, a sequence of temporal relations *m* is a *signature* in the sequence *w* of timed observations if and only if the anticipation rate T_A of *m* in *w* is greater than or equal to T_{Amin}. Thus, for anticipating the occurrence of an observation class C_i, a signature ending in C_i can be used as predictive model.

In some cases, although the anticipation rate T_A of a model *m* is above the value established T_{Amin} (i.e. *m* is a signature), the number of occurrences of *m* in *w* is

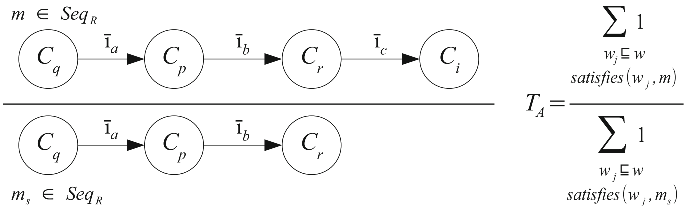


Fig. 6.11 Anticipation rate of m with regard to w (based on [13, p. 47])

low; or put in another way, the number of occurrences of the class C_i to be predicted is low. Therefore, in order to discard signatures where the occurrences of the last class are not significant in w, the coverage rate is established and illustrated in Fig. 6.12. Thus, the coverage rate T_C of m in w is the rate between the number of subsequences of w that satisfy m and the number of occurrences of the last observation class in m.

TOM4L aims, among other things, to discover from a given sequence w, a minimal set of signatures able to predict the maximal number of observations classes defined. That is, to discover a minimal set of temporal relations m whose anticipation rate T_A and coverage rate T_C in w are above the established threshold.

6.6.5 TOM4L Process

The general structure of the TOM4L KDD process is illustrated in Fig. 6.13 and is implemented by the ElpLab Java software, which allows to apply this data mining approach in an automatic way.

As depicted in the figure, stochastic and temporal properties of binary relations are obtained from a stochastic representation which associates a superposition of the Poisson process and a Markov chain with a set Ω of timed observations. A minimal set $R = \{r_j\}_{j=1,\ldots,r}$ of binary temporal relations which satisfies a criterion of interest is then induced from this stochastic representation, where the used criterion of interest is based on the BJ-measure described in Sect. 6.6.3.

From the mentioned minimal set, the TOM4L KDD process allows to compute a Naive Bayesian Network by means of the Tom4BN algorithm and a set of

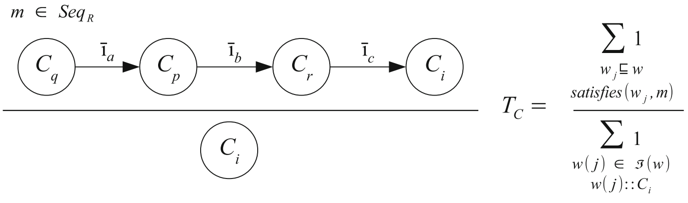


Fig. 6.12 Coverage rate of m with regard to w (based on [13, p. 47])

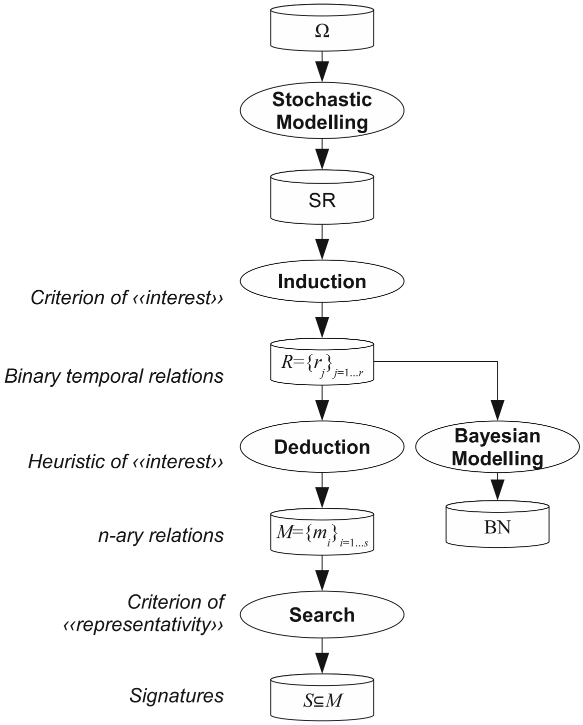

Ω
Stochastic Modelling
SR
Criterion of ‹‹interest››
Induction
Binary temporal relations
$R=\{r_j\}_{j=1\ldots r}$
Heuristic of ‹‹interest››
Deduction
Bayesian Modelling
n-ary relations
$M=\{m_i\}_{i=1\ldots s}$
BN
Criterion of ‹‹representativity››
Search
Signatures
$S\subseteq M$

Fig. 6.13 TOM4L KDD process [12, p. 40]

representative n-ary temporal relations. This latter is built through an abductive reasoning which is carried out on R in order to build a minimal set $M = \{m_i\}_{i=1,\ldots,s}$ of n-ary temporal relations m_i which would represent some properties of the process. In particular, the depth of abduction is controlled by heuristics based on the BJ-measure.

In the next stage, an exhaustive search is accomplished to extract from M the minimal set $S \subseteq M$ of n-ary temporal relations which satisfy a criterion of representativeness adapted to temporal relations. These n-ary relations are called signatures and their predictive ability allows to anticipate the occurrence of observable events. Searching for signatures consists in identifying all n-ary temporal relations m_i which finish in a particular observation class C_j (all paths and sub-paths to C_j) whose representativeness in Ω is sufficient. This representativeness is calculated from the coverage and the anticipation rates. The coverage rate of an n-ary relation m_i is the rate between the number of instances of m_i and the number of observation occurrences of class C_j; and, the anticipation rate of an n-ary relation m_i is the rate between the number of instances of the relation m_i and the number of instances of the relation m'_i, where m'_i is the result of removing the last observation class C_j of the path m_i.

Models obtained through the TOM4L process, both signatures and Bayesian Networks, can be related to TOM4D models as described in the next section.

6.7 Application to a Didactic Example

In this section, the proposed modelling approach combining TOM4D with TOM4L is described by means of a case study about the diagnosis of problems with a car. This case study has been taken from the book by Schreiber et al. [14] where it is presented by the authors in order to describe concepts and components of a CommonKADS Knowledge Model. Figure 6.14 depicts the domain knowledge of this case study where nine rules constitute the knowledge provided by an expert. These ones 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 then the engine does not start*, and so on.

From the introduced didactic problem, a summary description of applying TOM4D and TOM4L to this example, along with the relation between the obtained models, will be presented below. For the interested reader, the complete description of TOM4D modelling process applied to the example can be found in [4], and the detailed application of the TOM4L algorithms to the same example can be found in [8].

6.7.1 *TOM4D Models*

The interpretation of available knowledge requires an organization of this one. Thus, organizing and structuring knowledge is the first step in the modelling activity of the TOM4D KE methodology.

6.7.1.1 Organizing Available Knowledge

CommonKADS is an important methodology in terms of modelling experts' knowledge and therefore, it is utilized by TOM4D as a framework of interpretation and organization of knowledge. CommonKADS provides a collection of predefined

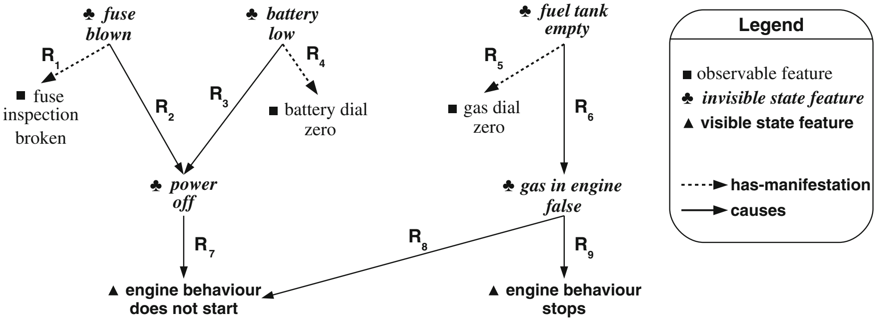


Fig. 6.14 Classification and organization of knowledge pieces

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. In this case, we shall consider the *diagnosis* template.

The diagnosis template presents a typical domain schema in which each system being diagnosed can be characterized 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, as Fig. 6.14 illustrates, 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 to represent 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.

Considering the previous classification, the arrows in Fig. 6.14 show dependences between the knowledge pieces. These dependences are rules which indicate relations between domain concepts. For example, “if there is no gas in engine, engine stops” establishes a causal relation between the concepts “gas in engine” and “engine behavior”: 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 other entity.

Thus, the previous reasoning, illustrated in Fig. 6.14, describing dependence types and concept types in the specific domain determines the following domain rules specified in (6.3) in the language CLM (Conceptual Modelling Language, [14]).

```
fuse.status = blown HAS-MANIFESTATION                           (R1)
              fuse-inspection.value = broken;
fuse.status = blown CAUSES power.status = off;                  (R2)
battery.status = low CAUSES power.status = off;                 (R3)
battery.status = low HAS-MANIFESTATION                          (R4)
              battery-dial.value = zero;
fuel-tank.status = empty HAS-MANIFESTATION                      (R5)
              gas-dial.value = zero;                                    (6.3)
fuel-tank.status = empty CAUSES                                 (R6)
              gas-in-engine.status = false;
power.status = off CAUSES                                       (R7)
              engine-behaviour.status = does-not-start;
gas-in-engine.status = false CAUSES                             (R8)
              engine-behaviour.status = does-not-start;
gas-in-engine.status = false CAUSES                             (R9)
              engine-behaviour.status = stops;
```

Considering the aforementioned analysis and the three TOM4D principles introduced in Sect. 6.5.3, the next objective is to define a scenario model $M(\Omega) = < SM(\Omega), FM(\Omega), BM(\Omega) >$ from the given knowledge and a set Ω of sequences of timed measures or observations 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, we shall deduce on the basis of the existing domain knowledge a scenario Ω to be assumed.

6.7.1.2 Knowledge Interpretation

The rules in (6.3) represent 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, that is, a scenario Ω. Taking in consideration $(R_1)$and (R_2) in (6.3), 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, it is supposed a scenario Ω of timed observations such that $\Omega = \{w_1, w_2, w_3, w_4\}$ where

$$\begin{aligned}
w_1 =& ((blown, t_{10}),\ (broken, t_{10} + \Delta t_{11}),\ (off, t_{10} + \Delta t_{11} + \Delta t_{12}),\\
& (does\_not\_start, t_{10} + \Delta t_{11} + \Delta t_{12} + \Delta t_{13}))\\
w_2 =& ((low, t_{20}),\ (battery\_zero, t_{20} + \Delta t_{21}),\ (off, t_{20} + \Delta t_{21} + \Delta t_{22}),\\
& (does\_not\_start, t_{20} + \Delta t_{21} + \Delta t_{22} + \Delta t_{23}))\\
w_3 =& ((empty, t_{30}),\ (gas\_zero, t_{30} + \Delta t_{31}),\ (false, t_{30} + \Delta t_{31} + \Delta t_{32})\\
& (does\_not\_start, t_{30} + \Delta t_{31} + \Delta t_{32} + \Delta t_{33}))\\
w_4 =& ((empty, t_{40}),\ (gas\_zero, t_{40} + \Delta t_{41}),\ (false, t_{40} + \Delta t_{41} + \Delta t_{42}),\\
& (stop, t_{40} + \Delta t_{41} + \Delta t_{42} + \Delta t_{43}))
\end{aligned} \tag{6.4}$$

From the interpretation of the available knowledge, the concepts fuse, battery, fuel-tank, battery-dial and gas-dial are considered as 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, engine-behaviour.status, etc. Thus, these variables and components are defined in (6.5) where the value that, in principle, a variable x_i $(i = 1, \ldots, 9)$ can assume, is described in the corresponding set Δ_{x_i} presented in (6.6), denoting ϕ_i an unknown value.

$$\begin{array}{ll}
\text{Variables}\, X = \{x_1, \ldots, x_9\} & \text{Components}\, COMPS = \{c_1, \ldots, c_9\} \\
x_1 \triangleq \texttt{fuse.status} & c_1 \triangleq \texttt{fuse} \\
x_2 \triangleq \texttt{battery.status} & c_2 \triangleq \texttt{battery} \\
x_3 \triangleq \texttt{fuel}-\texttt{tank.status} & c_3 \triangleq \texttt{fuel}-\texttt{tank} \\
x_4 \triangleq \texttt{fuse}-\texttt{inspection.value} & c_4 \triangleq \texttt{fuseinspectiontools} \\
x_5 \triangleq \texttt{battery}-\texttt{dial.value} & c_5 \triangleq \texttt{battery}-\texttt{dial} \\
x_6 \triangleq \texttt{gas}-\texttt{dial.value} & c_6 \triangleq \texttt{gas}-\texttt{dial} \\
x_7 \triangleq \texttt{power.status} & c_7 \triangleq \texttt{electricsupply} \\
x_8 \triangleq \texttt{gas}-\texttt{in}-\texttt{engine.status} & c_8 \triangleq \texttt{gassupply} \\
x_9 \triangleq \texttt{engine}-\texttt{behaviour.status} & c_9 \triangleq \texttt{engine}
\end{array} \tag{6.5}$$

$$\begin{array}{lll}
\Delta_{x_1} = \{blown, \phi_1\} & \Delta_{x_4} = \{broken, \phi_4\} & \Delta_{x_7} = \{off, \phi_7\} \\
\Delta_{x_2} = \{low, \phi_2\} & \Delta_{x_5} = \{battery\_zero, \phi_5\} & \Delta_{x_8} = \{false, \phi_8\} \\
\Delta_{x_3} = \{empty, \phi_3\} & \Delta_{x_6} = \{gas\_zero, \phi_6\} & \Delta_{x_9} = \{stops,\ does\_not\_start\}
\end{array} \tag{6.6}$$

In the first phase the scenario model $M(\Omega) = {<}SM(\Omega), FM(\Omega), BM(\Omega){>}$ is defined. Although the detailed specification of this model will not be presented, we shall mention the principal points of this one. This model organizes and describes the information and the knowledge available. $SM(\Omega)$ describes the 9 components in (6.5) and the interconnections between them; and $FM(\Omega)$ specifies the relation among the values that the variables can assume through the definition of a set of functions. For example, rule R_5 allows to establish an interconnection between the components c_3 (fuel-tank) and c_6 (gas-dial); and also, the relation between the values of x_3 and x_6 through a function $f_1 : \Delta_{x_3} \to \Delta_{x_6}$ such that $f_1(empty) = gas\_zero, f_1(\phi_3) = \phi_6$, and where $x_6 = f_1(x_3)$. Besides, $BM(\Omega)$ specifies an initial behavioural model that, because of the 9 existent binary variables, consists of 18 observation classes (e.g., $C_{1,1} = \{(x_1, blown)\}$, $C_{1,2} = \{(x_1, \phi_1)\}$ are observation classes related to x_1) and $2^9 = 512$ characterized states (e.g., a state in which $x_1 = blown,\ x_2 = low,\ x_3 = empty,\ x_4 = \phi_4,\ x_5 = battery\_zero,\ x_6 = gas\_zero,\ x_7 = off,\ x_8 = false,\ x_9 = stops$). 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 discernible states, a number significant with respect to the small number of units. Presumably, certain states in $BM(\Omega)$ are irrelevant for the pursued objectives or, they are meaningless since are impossible physically. Then, the two following stages in the modelling process, illustrated in Fig. 6.8, Sect. 6.5.3, aim to deal with these aspects.

6.7.1.3 Process Definition

In the phase of process definition, the perception model $PM(X(t))$ is defined, where the boundaries and operating constraints such as the set of variables of interest, operating goals, normal and abnormal operating modes are established. After that, in the stage of generic modelling, the objective is to define a model already not of a particular scenario, but a more general model of the car functioning. These two stages resort to the Formal Logic and the Tetrahedron of States in order to carry out a logical and a physical interpretation of the variables as Table 6.1 describes.

From Formal Logic, the components c_i $(i = 1,\ldots,9)$ in (6.5) can be considered as logical components c_{Bi} described with first order predicate logic where Reiter's diagnosis theory [44] can be applied. Thus, the variables x_i $(i = 1,\ldots,9)$ can be interpreted as logical variables $\bar{x}_i$ $(i = 1,\ldots,9)$ which assume values 1 (true) or 0 (false). For example, in Table 6.1, $x_1 = blown$ is logically interpreted as $\bar{x}_1 = 0$ (false).

In principle, the components c_4 (*fuse inspection tools*), c_5 (battery-dial) and c_6 (*gas-dial*) being sensors, they simply replicates the behaviour of the components c_1 (fuse), c_2 (battery) and c_3 (fuel-tank). Consequently, and for reducing the complexity, it is assumed that the former work correctly (i.e. sensors are supposed to never fail) and then they are not necessary in the resultant model. Thus, the logical model of the process is depicted in Fig. 6.15 and the logical relations

Table 6.1 Logical and physical interpretations

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

[a] $\alpha \in \{0, 1\}$ models the car key (off/on) allowing to interpret $x_9 = \neg works$ as the car is stopped (owing to 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 can not be observed

among the variables are presented in Table 6.2. In the figure, the boxes c_{B7}, c_{B8} and c_{B9} represent logical "AND" components, and the components c_{B1}, c_{B2} and c_{B3} represent boolean value generators. This interpretation allows to specify clearly conditions of normal and abnormal behaviour on the variables and, as mentioned, it allows to resort to Reiter's theory.

Nevertheless, 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. For example, when observing Fig. 6.15, a state in which $\bar{x}_3 = 0$ and $\bar{x}_8 = 1$ results in an *inconsistent* state and in the mentioned theory, this would indicate that the component c_{B8} does not work. However, in this *inconsistent* state, the fuel tank is empty $(\bar{x}_3 = 0)$ and there is gas in the engine $(\bar{x}_8 = 1)$; consequently, the mentioned situation can not be associated with the problem of a component. On the contrary, this state is transient and corresponds to *normal* behaviour, although it is not a state of interest for the diagnosis task and it should not be considered. However, in the logical model, it is identified as a state of abnormal behaviour.

The example shows that the logical interpretation of variables required by Reiter's theory must be completed with a physical interpretation. For this purpose, [40] proposes to utilize the Tetrahedron of States (ToS), introduced in Sect. 6.5.2, where the given variables can be mapped to physical variables of the ToS and thus, the relations among them established. 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. 6.16a, b, are used in this example.

Each given variable $x_i \in X$ is mapped to a physical variable of the corresponding ToS. For example, using the Hydraulic ToS in Fig. 6.16a, the variable x_3 (fuel tank status) is associated with the gas volume $V(t)$ in the tank, as Table 6.1 specifies where $V(t)$ is also noted as x_3^p. Thus, $x_3 = empty$ which is logically interpreted as $\bar{x}_3 = 0$ (false), is physically interpreted through the ToS as $V(t) = 0$; and, $x_3 = \neg empty$ (or $x_3 = \phi_3$), related to $\bar{x}_3 = 1$ (true), is physically interpreted as $V(t) \neq 0$.

Table 6.2 Logical and physical functional relations

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

Thereby, the variables are mapped with physical variables as illustrated in Fig. 6.16 and specified in Table 6.1 where the relations among the variables are established as Table 6.2 presents. Thus, the physical model of the process is illustrated in Fig. 6.17.

This interpretation allows to determine conditions on the variables in order to identify transient states which can be discarded from the model to be built. For example, the states in which $V(t) = 0$ and $Qv(t) \neq 0$ (see Fig. 6.17) can be eliminated from the model; or, what is the same thing, the states in which $\bar{x}_3 = 0 \wedge \bar{x}_8 = 1$.

From this interpretation and a suitable analysis, the transient or physically impossible states can be removed from the model to be built, which results in 21 states of interest.

6.7.1.4 Generic Modelling

From the consideration of the two previous interpretations, a generic model of the process $M(X(t)) = <PM(X(t)), SM(X(t)), FM(X(t)), BM(X(t))>$ is defined. Details of the analysis carried out for establishing this model will not be described. Nevertheless, we shall limit ourselves to present the resultant model of the process formally specified in the TOM4D formalism [4].

In order to facilitate the analysis, the logical variables are used to describe the model; considering always that it is possible to reinterpret them like the variables and components described in (6.5) through Table 6.1. Thereby, observing Fig. 6.15, the models are the following ones.

The perception model $PM(X(t))$ of the process consists of the set X of variables, the set Ψ of threshold values described in Sect. 6.4 which in this case are not present and a set R_q of sentences describing objectives and operating modes. This model is specified as follows:

$$\begin{aligned}
&PM(X(t)) = <X, \Psi, R_q> \text{ where}\\
&\quad X = \{\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_7, \bar{x}_8, \bar{x}_9\}, \quad \Delta_{\bar{x}_i} = \{0,1\}, \quad i = 1,2,3,7,8,9\\
&\quad \Psi = \{\Psi_i\}_{i=1,2,3,7,8,9} \quad \text{(threshold values of the time functions}\\
&\qquad\qquad\qquad\qquad \text{which we do not know)}\\
&\quad R_q = R_{goal} \cup R_n \cup R_{ab} \text{ such that}\\
&\quad R_{goal} \text{ describes the process operating goals } \bar{x}_9 = 1\\
&\quad R_n \text{ describes the conditions of the normal operating mode :}\\
&\quad (\bar{x}_1 = 1 \wedge \bar{x}_2 = 1 \wedge \bar{x}_3 = 1 \wedge \bar{x}_7 = 1 \wedge \bar{x}_8 = 1 \wedge \bar{x}_9 = 1) \vee\\
&\quad ((((\bar{x}_1 = 0 \vee \bar{x}_2 = 0) \wedge \bar{x}_7 = 0) \vee (\bar{x}_3 = 0 \wedge \bar{x}_8 = 0)) \wedge \bar{x}_9 = 0)\\
&\quad R_{ab} \text{ describes the conditions of the abnormal operating mode :}\\
&\quad (\bar{x}_1 = 1 \wedge \bar{x}_2 = 1 \wedge \bar{x}_7 = 0) \vee (\bar{x}_3 = 1 \wedge \bar{x}_8 = 0) \vee (\bar{x}_7 = 1 \wedge \bar{x}_8 = 1 \wedge \bar{x}_9 = 0)
\end{aligned} \tag{6.7}$$

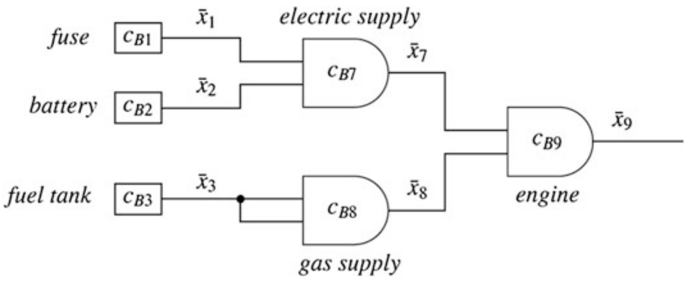


Fig. 6.15 Logical model of the process

The structural model $SM(X(t))$, defined in (6.8), describes the set $COMPS$ of components, the set R_{port} specifying the interconnections between output ports with input ports of components (e.g., $out(c_{B1}) = in_1(c_{B7})$) and the set R_{xport} associating each variable with an output port (e.g., $out(c_{B1}) = \bar{x}_1$).

$$\begin{aligned} SM(X(t)) = \; & <COMPS, R_{port}, R_{xport}> \text{ where} \\ & COMPS = \{c_{B1}, c_{B2}, c_{B3}, c_{B7}, c_{B8}, c_{B9}\} \\ & R_{port} = \{out(c_{B1}) = in_1(c_{B7}), out(c_{B2}) = in_2(c_{B7}), out(c_{B7}) = in_1(c_{B9}), \\ & \qquad out(c_{B3}) = in_1(c_{B8}), out(c_{B3}) = in_2(c_{B8}), out(c_{B8}) = in_2(c_{B9})\} \\ & R_{xport} = \{out(c_{B1}) = \bar{x}_1, out(c_{B2}) = \bar{x}_2, out(c_{B3}) = \bar{x}_3, \\ & \qquad out(c_{B7}) = \bar{x}_7, out(c_{B8}) = \bar{x}_8, out(c_{B9}) = \bar{x}_9\} \end{aligned} \tag{6.8}$$

The functional model $FM(X(t))$ describes the relations among the values that the variables can assume, as defined in (6.9). This model consists of the set Δ of values belonging to the domain and the image of the functions defined in the set F, the mentioned set F and the set R_f that establishes the relation among the variables (e.g., $\bar{x}_7 = f_{B4}(\bar{x}_1, \bar{x}_2)$).

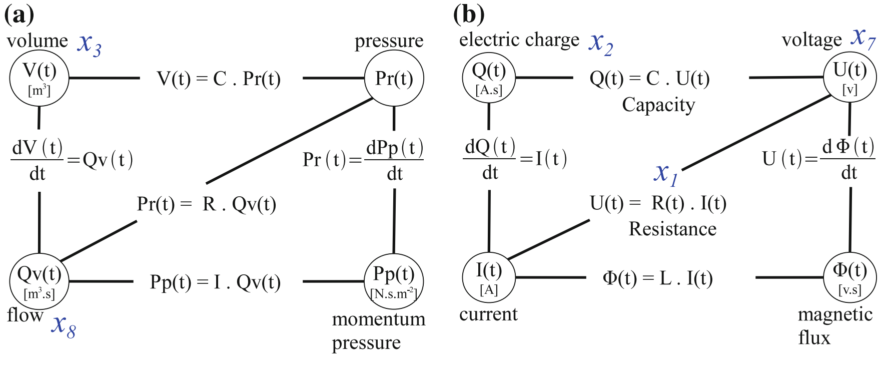
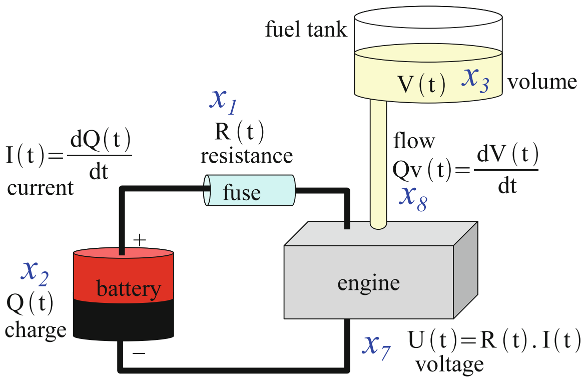


Fig. 6.16 Physical interpretation of variables

Fig. 6.17 Physical model of the process



$$FM(X(t))$$

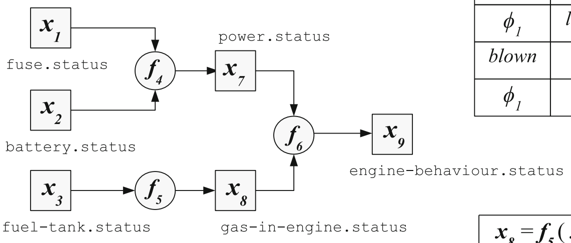
$$\begin{aligned}
&\Delta_{\bar{x}_i} = \{0,1\}, \quad i = 1,2,3,7,8,9\\
&F = \{f_{B4}, f_{B5}, f_{B6}\} \text{ with}\\
&f_{B4} : \Delta_{\bar{x}_1} \times \Delta_{\bar{x}_2} \to \Delta_{\bar{x}_7}\\
&f_{B5} : \Delta_{\bar{x}_3} \to \Delta_{\bar{x}_8}\\
&f_{B6} : \Delta_{\bar{x}_7} \times \Delta_{\bar{x}_8} \to \Delta_{\bar{x}_9} \quad \text{and such that}\\
&\quad f_{B4}(y_1,y_2) = and\ (y_1,y_2) \wedge f_{B5}(y) = and(y,y) \wedge\\
&\quad f_{B6}(y_1,y_2) = and\ (y_1,y_2)\\
&R_f = \{\bar{x}_7 = f_{B4}(\bar{x}_1,\bar{x}_2),\ \ \bar{x}_8 = f_{B5}(\bar{x}_3),\ \ \bar{x}_9 = f_{B6}(\bar{x}_7,\bar{x}_8)\}
\end{aligned} \tag{6.9}$$

For readability and clarity, we consider to reinterpret from Table 6.1 the logical variables $\bar{x}_i$ ($i = 1,2,3,7,8,9$) like their corresponding x_i ($i = 1,2,3,7,8,9$). This reinterpretation then allows us to see the functional model as depicted in Fig. 6.18.

The behavioural model requires the set of observation classes, which is defined as $C = \{C_{1,1}, C_{1,2}, C_{2,1}, C_{2,2}, C_{3,1}, C_{3,2}, C_{7,1}, C_{7,2}, C_{8,1}, C_{8,2}, C_{9,1}, C_{9,2}\}$ where

$$\begin{array}{llll}
C_{1,1} = \{(\bar{x}_1,0)\}, & C_{2,2} = \{(\bar{x}_2,1)\}, & C_{7,1} = \{(\bar{x}_7,0)\}, & C_{8,2} = \{(\bar{x}_8,1)\},\\
C_{1,2} = \{(\bar{x}_1,1)\}, & C_{3,1} = \{(\bar{x}_3,0)\}, & C_{7,2} = \{(\bar{x}_7,1)\}, & C_{9,1} = \{(\bar{x}_9,0)\},\\
C_{2,1} = \{(\bar{x}_2,0)\}, & C_{3,2} = \{(\bar{x}_3,1)\}, & C_{8,1} = \{(\bar{x}_8,0)\}, & C_{9,2} = \{(\bar{x}_9,1)\}
\end{array} \tag{6.10}$$

From this set and the a priori knowledge, the possible sequences of observations classes are defined, as Fig. 6.19 depicts; that is, it is considered possible that after an occurrence of the class $C_{1,1}$ (i.e., the fuse is blown) an occurrence of the class $C_{7,1}$ (the power is off) is observed, then the sequential relation $(C_{1,1}, C_{7,1})$ is present in the figure.



$x_7 = f_4(x_1, x_2)$

x_1	x_2	x_7
blown	*low*	*off*
ϕ_1	*low*	*off*
blown	ϕ_2	*off*
ϕ_1	ϕ_2	ϕ_7

$x_9 = f_6(x_7, x_8)$

x_7	x_8	x_9
off	*false*	¬ *works*
off	ϕ_8	¬ *works*
ϕ_7	*false*	¬ *works*
ϕ_7	ϕ_8	ϕ_9

$x_8 = f_5(x_3)$

x_3	x_8
empty	*false*
ϕ_3	ϕ_8

Fig. 6.18 Functional model

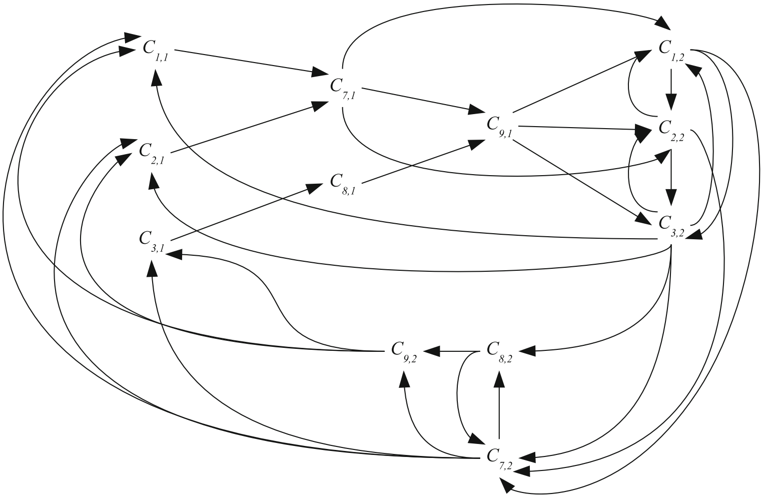


Fig. 6.19 Graphical representation of the possible sequences of observation classes

The occurrence of an observation class entails the assignation of a value to a variable; that is to say, an occurrence of $C_{1,1}$ entails that the value 0 is assumed by $\bar{x}_1$. Consequently, the previous value of $\bar{x}_1$ was not 0. Thus, the possible states between two observation classes can be characterized and established. Recall that only 21 characterized states were considered of interest from the logical and physical interpretations.

Thereby, the behavioural model $BM(X(t))$, defined in (6.11) and illustrated in Fig. 6.20, consists of the set S of characterized states, the set C of observation classes and the transition function γ.

$$BM(X(t)) = <S, C, \gamma> \quad \text{where}$$

$$\begin{aligned} S = & \{s_8, s_{11}, s_{17}, s_{18}, s_{20}, s_{21}, s_{23}, s_{24}, s_{27}, s_{28}, s_{29}, s_{30}, s_{31}, s_{32}, \\ & s_{50}, s_{53}, s_{56}, s_{61}, s_{62}, s_{63}, s_{64}\} \quad \text{such that} \end{aligned} \tag{6.11}$$

$$S = \{s : VAR \rightarrow VALUE | \\ s(x) = \delta, x \in X \subseteq VAR, \delta \in \Delta \subseteq VALUE\}$$

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_{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

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

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

$$\gamma : S \times C \rightarrow S \quad \text{such that}$$

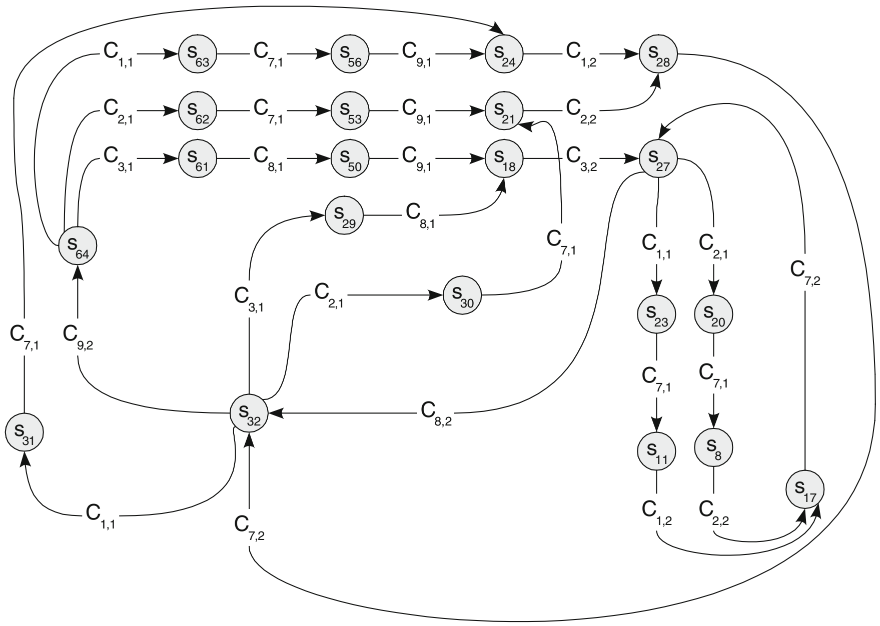


Fig. 6.20 Behavioural model of the process P(t)

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

As a result of this analysis, 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. These interpretations along with modelling decisions allowed a reduction from 512 to only 21 states physically possible and of interest for diagnosing behaviour problems. The logical model of Fig. 6.15 describes the structure of the expert's diagnosis reasoning and the physical model of Fig. 6.17 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, introduce the dimension of time allowing to model the dynamic of the process in a behavioural model. This model is a crucial element in the supervision of processes since generally it is collated with the real process evolution. This quadripartite structure of the model discriminates the different types of knowledge about the process and then allows greater understanding of the problem and better communication with experts.

6.7.2 TOM4L Models

The models described in this section have been automatically provided by the ElpLab Java software which implements the complete TOM4L KDD process as illustrated in Fig. 6.13, Sect. 6.6.5. For this purpose, based on the scenario Ω defined in (6.4), Sect. 6.7.1.2, and from the method described in [49], a set of 100 occurrences of the observation classes $C_{1,1}$, $C_{2,1}$, $C_{3,1}$, $C_{7,1}$, $C_{8,1}$ and $C_{9,1}$, with a stochastic distribution of time according to Table 6.3, was built.

As described in Sect. 6.6, the TOM4L learning approach groups data mining algorithms and techniques which provide the possibility of finding n-ary temporal relations among observation classes in timed data. Thus, from the sequence w which is made up the 100 timed observations, a Functional Model and a Behavioural Model of the car functioning can be obtained when applying TOM4L.

6.7.2.1 Functional Model

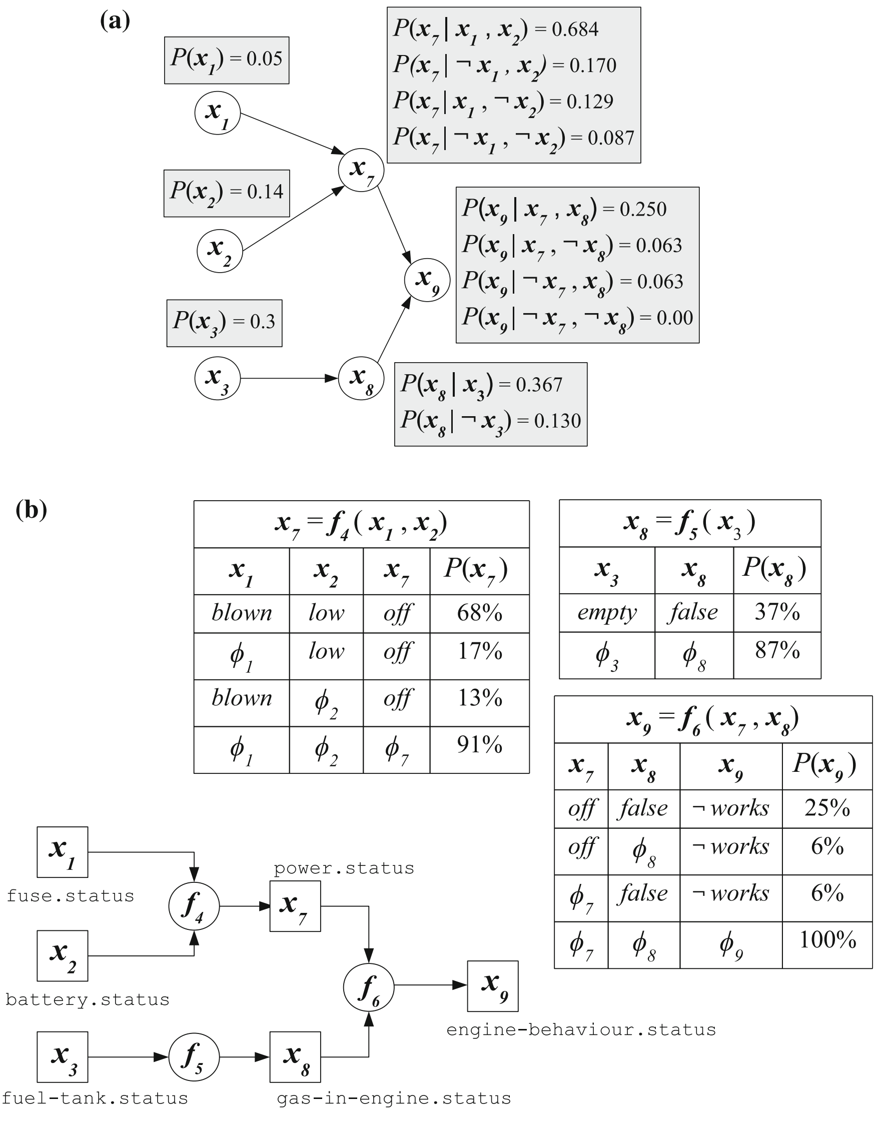
The algorithm Tom4BN [8, 10] which allows to discover naive Bayesian Networks (Sect. 6.6.3) from timed data is applied to the 100 observations of the car example, giving as a result the Bayesian Network shown in Fig. 6.21a.

In this example, classes $C_{i,j}$ are singletons of the form $C_{i,j} = \{(x_i, \delta_j)\}$ and $P(C_{i,j})$, equivalent to $P(x_i = \delta_j)$, is the prior probability of observing an occurrence of the class $C_{i,j} = \{(x_i, \delta_j)\}$ in w. Besides, it should be noted that "$\neg x_i$" refers to any equality except "$x_i = \delta_j$" or, put in another way, "$\neg x_i$" denotes "$x_i = \delta_k \wedge \delta_k \neq \delta_j$".

Thus, this Bayesian Network enables the definition of the Functional Model of Fig. 6.21b, whose functions correspond to those ones of the TOM4D Functional Model (Fig. 6.18, Sect. 6.7.1.4); but unlike the last one, these functions have probabilities associated which provide a certain level of confidence about the established relations among values. For example, the probability of observing that the power is off having observed that the battery is low and the fuse is blown is 0.684; that is, $P(x_7 | x_1, x_2) = 0.684$ in Fig. 6.21a. Thus, the level of confidence

Table 6.3 Prior probabilities of the car example [10, p. 76]

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


(a)
$P(x_1) = 0.05$
x_1
$P(x_2) = 0.14$
x_2
$P(x_3) = 0.3$
x_3
$P(x_7 | x_1, x_2) = 0.684$
$P(x_7 | \neg x_1, x_2) = 0.170$
$P(x_7 | x_1, \neg x_2) = 0.129$
$P(x_7 | \neg x_1, \neg x_2) = 0.087$
x_7
$P(x_9 | x_7, x_8) = 0.250$
$P(x_9 | x_7, \neg x_8) = 0.063$
$P(x_9 | \neg x_7, x_8) = 0.063$
$P(x_9 | \neg x_7, \neg x_8) = 0.00$
x_9
x_8
$P(x_8 | x_3) = 0.367$
$P(x_8 | \neg x_3) = 0.130$
(b)
x_1
fuse.status
x_2
battery.status
x_3
fuel-tank.status
f_4
f_5
power.status
x_7
x_8
gas-in-engine.status
f_6
x_9
engine-behaviour.status

$x_7 = f_4(x_1, x_2)$			
x_1	x_2	x_7	$P(x_7)$
blown	*low*	*off*	68%
ϕ_1	*low*	*off*	17%
blown	ϕ_2	*off*	13%
ϕ_1	ϕ_2	ϕ_7	91%

$x_8 = f_5(x_3)$		
x_3	x_8	$P(x_8)$
empty	*false*	37%
ϕ_3	ϕ_8	87%

$x_9 = f_6(x_7, x_8)$			
x_7	x_8	x_9	$P(x_9)$
off	*false*	¬ *works*	25%
off	ϕ_8	¬ *works*	6%
ϕ_7	*false*	¬ *works*	6%
ϕ_7	ϕ_8	ϕ_9	100%

Fig. 6.21 Functional model obtained through TOM4L [10, pp. 81, 83]

when considering $off = f_4(blown, low)$ is approximately 68 % as Fig. 6.21b depicts. Another example is the probability of $\phi_7 = f_4(\phi_1, \phi_2)$, which can be obtained from $P(x_7|\neg x_1, \neg x_2) = 0.087$ when calculating $P(\neg x_7|\neg x_1, \neg x_2) = 1 - P(x_7|\neg x_1, \neg x_2)$.

Hence, the Functional Model with probabilities automatically obtained from data can be compared with the Functional Model defined from experts' knowledge; and thus, both models can be analysed together complementing each other.

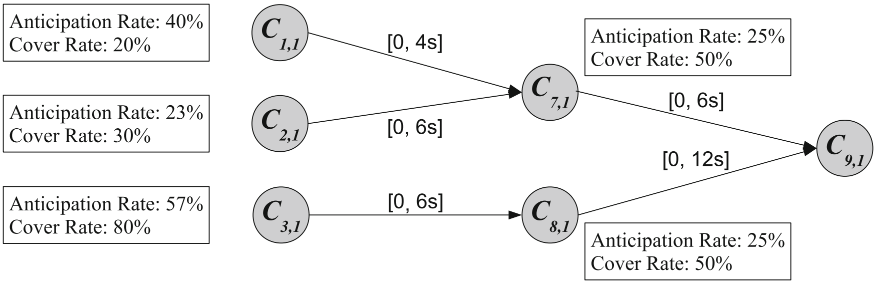


Fig. 6.22 Behavioural model obtained through TOM4L. Signature tree of the observation class $C_{9,1}$ [10]

6.7.2.2 Behavioural Model

A behavioural model can also be obtained from timed data through the TOM4L process. The algorithm BJT4S [9] is applied to the set of observation sequences, and consequently, the model in Fig. 6.22 is automatically obtained.

The figure presents the sequences of observation classes discovered from data where the values between brackets denote the average maximum and minimum time periods between two occurrences of observation classes; that is, the temporal constraints as described in Sect. 6.6.1. This model is a tree whose branches (called *signatures* [13, 49] and described in Sect. 6.6.4) define n-ary temporal relations among observation classes and verify certain anticipation and coverage rates. For example, as shown in Fig. 6.22, $m = ((C_{1,1}, C_{7,1}, [0, 4s]), (C_{7,1}, C_{9,1}, [0, 6s]))$ is a signature which denotes the sequence of the type $C_{1,1}, C_{7,1}, C_{9,1}$ with its temporal constraints. In the figure, the anticipation rate of the mentioned signature m 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 $4s$, then an occurrence of $C_{9,1}$ takes place in at most $6s$. For its part, its coverage rate means that in 20 % of the cases in which an occurrence of $C_{9,1}$ is observed, the signature $m = ((C_{1,1}, C_{7,1}, [0, 4s]), (C_{7,1}, C_{9,1}, [0, 6s]))$ was verified.

Clearly, this model is a sub-model of that one in Fig. 6.19, Sect. 6.7.1.4, describing sequences of observation classes built through TOM4D. Therefore, the model of Fig. 6.22 implicitly determines a behavioural model which is included in the TOM4D Behavioural Model defined from experts' knowledge (Fig. 6.20, Sect. 6.7.1.4). In particular, the model obtained from data provides, in addition, knowledge about temporal constraints between event occurrences. Thus, once again, these models belonging to different disciplines, such as KE and KDD are, can be easily related and compared to each other.

Owing to that, TOM4L models can be related with TOM4D models and the latter are directly related with a CommonKADS conceptual model, the communication with experts about the first one is easier. That is to say, the meaning of the signature $m = ((C_{1,1}, C_{7,1}, [0, 4s]), (C_{7,1}, C_{9,1}, [0, 6s]))$ can be easily explained by saying that in the 40 % of cases, when observing the fuse blown, in at most 4*s* the power is observed off and subsequently, in at most 6*s*, it is observed that the engine does not work. Thus, TOM4D establishes a bridge between experts' knowledge and data, and TOM4L allows automatic learning from these last ones.

6.8 Conclusion

Knowledge acquisition, as a topic of interest in sciences, has been generally addressed from two different perspectives. One approach has been to consider knowledge acquisition as a psychological and social process that consists in the synthesis of new knowledge through socialization with experts. The other approach has been to consider knowledge acquisition as an interpretation and analysis process of data, based on discovering patterns of interest through observation, analysis and intertwining of the data. These two perspectives are, respectively, central issues in Knowledge Engineering (KE) and in Knowledge Discovery in Database (KDD).

Nevertheless, as highlighted by N. Wickramasinghe [15], although knowledge acquisition is the main and central question in both disciplines, the issue has been traditionally approached from one or the other perspective, rather than from an integrative view. We consider then that a whole approach is necessary in order to accelerate the global learning process and even, in extremely complex cases, to provide viability.

Results about probabilistic information and temporal constraints, as well as discovered event sequences which could be unexpected, extend the knowledge about a real process and provide resources to build a more suitable model of this one. However, relating this knowledge to the expert's one is not a trivial task because, generally, the formalisms used by Knowledge Engineering methodologies and by Knowledge Discovery in Database processes to represent knowledge models are different. As a consequence, the comparison between both models can not be *in principle* carried out. We then believe that the main difficulty for relating the mentioned disciplines stems from the lack of a global approach based on a same theory and consequently, from the lack of representation formalisms that can be used in both domains.

Thereby, the central focus of this chapter was the definition of a global human–machine learning process which combines a Knowledge Engineering methodology called TOM4D (i.e. Timed Observation Modelling for Diagnosis) with a Knowledge Discovery in Database process called TOM4L (i.e. Timed Observation Mining for Learning). Thus, with the aim of defining this integral view, the Theory of Timed Observations [1] has been established as a basis for the development of the proposed approach. This theory defines, among other things, the notions of *timed observation* and *observation class*, concepts that enable to specify the traditional notion of discrete event and the Artificial Intelligence notion of alarm (or warning).

This chapter presented then the TOM4D Knowledge Engineering methodology, which allows to build models, by basing on the Theory of Timed Observations, from experts' knowledge. The models built through this methodology are not experts' Knowledge Models but models of the process about which experts have knowledge. By construction, TOM4D models are consistent with and easily relatable to CommonKADS Knowledge Models built from experts' knowledge, CommonKADS being one of the principal KE methodologies. Therefore, models of a process built through TOM4D facilitate the communication with the expert, and thus, the validation of the Knowledge Models. Besides, the chapter introduced the basic elements of the TOM4L Knowledge Discovery in Database process to obtain knowledge from data. The TOM4L process allows to find n-ary temporal relations of observation classes representative of the process that gives rise to data, by using an entropy-based measure called the BJ-measure [9, 12]. In addition, through the aforesaid measure, TOM4L enables to build Bayesian Networks from timed data [8, 10, 11]. Thus, TOM4L models are directly relatable to TOM4D models.

In summary, it was presented a human–machine learning process nourished from experts' knowledge and knowledge discovered in data which, in our opinion, is ultimately a virtuous circle that establishes a positive and corrective feedback to each step. Therefore, a process model which meets the expectation in the knowledge intensive tasks performed by a Knowledge Based System can be built in a more suitable way.

Real world problems have been addressed though this approach. In particular, the security of the dam of Cublize (France), where the resultant models have been validated by the hydraulic dam experts of the French governmental organization (Irstea) which controls the security of hydraulic civil engineering structures in the corresponding country [6, 50]. Moreover, nowadays we are utilizing the presented approach in order to model human behaviour from gerontologists' knowledge and smart environments data, in the context of the GerHome Project of the Centre Scientifique et Technique du Bâtiment (CSTB) of Sophia Antipolis, France [4, 51].

We believe that binding the KE and KDD universes enriches and facilitates the modelling task. Nevertheless, there still exists a difficulty with regard to the discursive and conceptual levels in which each universe is developed. That to say, sometimes, even being able to link the mentioned disciplines, relating models obtained from knowledge discovered in data to models obtained from experts' knowledge is very difficult, because experts' conceptual abstraction level is very high or is far from those concepts at data level. Although this topic has been beyond the scope of the present chapter, we consider of interest to mention that this issue has been addressed by means of a theoretical framework of abstraction levels that we have defined [4, 52, 53], where in each level a KE methodology, like TOM4D, can be combined with a KDD process, like TOM4L, in order to built a set of models linking the data abstraction level (e.g. sensor level) to the expert's conceptual level.

References

1. Le Goc, M.: 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)
2. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(379–423), 623–656 (1948)
3. Dagues, P.: Théorie logique du diagnostic à base de modèles. Diagnostic, Intelligence Artificielle, et Reconnaissance des Formes, pp. 17–105. Hermes Science Publications, Paris (2001)
4. Pomponio, L.: Definition of a human-machine learning process from timed observations: application to the modelling behaviour of old people at home. Université Aix-Marseille (2012)
5. Pomponio, L., Le Goc, M.: Timed observations modelling for diagnosis methodology: a case study. In: Cordeiro, J.A.M., Virvou, M., Shishkov, B. (eds.) ICSoft 2010—Proceedings of the 5th International Conference on Software and Data Technologies, pp. 504–507. SciTePress, Athens (2010)
6. Le Goc M., Masse E., Curt C.: Modeling processes from timed observations. In: Proceedings of the 3rd International Conference on Software and Data Technologies (ICSoft'08), pp. 249–256 (2008)
7. Le Goc, M., Masse, E.: Towards a multimodeling approach of dynamic systems for diagnosis. In: Proceedings of the 2nd International Conference on Software and Data Technologies (ICSoft'07), pp. 277–282 (2007)
8. Le Goc, M., Ahdab, A.: Learning Bayesian Networks from Timed Observations. LAP LAMBERT Academic Publishing GmbH & Co, KG (2012)
9. Benayadi, N., Le Goc, M.: Mining timed sequences with TOM4L framework. In: Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS 2010), pp. 111–120 (2010)
10. Ahdab, A., Le Goc, M.: Learning dynamic bayesian networks with the TOM4L process. In: Proceedings of the 5th International Conference on Software and Data Technologies (ICSoft 2010), pp. 353–363 (2010)
11. Ahdab, A.: Contribution à l'apprendissage de réseaux bayésiens à partir de donnèes datées pour le diagnostic des processus dynamiques continus. Université Paul Cézanne, Aix-Marseille (2010)
12. Benayadi, N.: Contribution à la découverte de connaissances à partir de données datées. Université Paul Cézanne, Aix-Marseille III (2010)
13. Bouché, P.: Une approache stochastique de modélisation de séquences d'événements discrets pour le diagnostic des systèmes dynamiques. Université Paul Cézanne, Aix-Marseille III (2005)
14. Schreiber, G., Akkermans, H., Anjewierden, A., et al.: Knowledge Engineering and Management: the CommonKADS Methodology. MIT Press, Cambridge (2000)
15. Wickramasinghe, N.: Knowledge Creation. Encyclopedia of Knowledge Management, pp. 326–335. Idea Group Inc., Hershey (2006)
16. Nonaka, I.: Dynamic theory of organizational knowledge creation. Organ. Sci. **5**, 14–37 (1994)
17. Nonaka, I.: The knowledge-creating company. Harvard Bus. Rev. 96–104 (1991)
18. Alavi, M., Leidner, D.E.: Review: knowledge management and knowledge management systems: conceptual foundations and research issues. MIS Quart **25**, 107–136 (2001)
19. Polanyi, M.: The Tacit Dimension. Doubleday & Company, Inc., NY (1966)
20. Nonaka, I., Konno, N.: The concept of "Ba": building a foundation for knowledge creation. California Manage. Rev. **40**, 40–54 (1998)

21. Feigenbaum, E.A.: The art of artificial intelligence: 1. Themes and case studies of knowledge engineering. In: International Joint Conference on Artificial Intelligence, pp. 1014–1029 (1977)
22. Feigenbaum, E.A.: A personal view of expert systems: looking back and looking ahead. knowledge systems laboratory. Department of Computer Science, Stanford University (1992)
23. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge Engineering: Principles and Methods. Data Knowl. Eng. **25**, 161–197 (1998)
24. Breuker, J., de Velde, W.V.: CommonKADS Library For Expertise Modelling. IOS Press, Amsterdam (1994)
25. Gennari, J.H., Musen, M.A., Fergerson, R.W., et al.: The evolution of protégé: an environment for knowledge-based systems development. Int. J. Hum Comput Stud. **58**, 89–123 (2002)
26. Angele, J., Fensel, D., Landes, D., Studer, R.: Developing knowledge based-systems with MIKE. Autom. Soft. Eng. **5**, 389–418 (1998)
27. Angele, J., Fensel, D., Studer, R.: Domain and task modeling in MIKE. In: Proceedings of the IFIP WG8.1/13.2 Joint Working Conference on Domain Knowledge for Interactive System Design, pp. 8–10 (1996)
28. Cairó, O., Alvarez, J.C.: KAMET II: an extended knowledge-acquisition methodology. In: Palade, V., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based Intelligent Information and Engineering Systems, pp. 61–67. Springer, London (2003)
29. Cairó, O., Alvarez, J.C.: The KAMET II Methodology: A Modern Approach for Building Diagnosis-Specialized Knowledge-Based Systems ISMIS, pp. 652–656. Springer, London (2003)
30. Motta, E., Stutt, A., O'Hara, K. et al.: VITAL knowledge representation language specification. Human Cognition Research Laboratory of the Open University (1991)
31. Piatetsky-Shapiro, G.: Knowledge discovery in real databases: a report on the IJCAI-89 workshop. IA Mag. **11**, 68–70 (1990)
32. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. IA Mag. **17**, 37–57 (1996)
33. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The KDD process for extracting useful knowledge from volumes of data. Commun. ACM **39**, 29–34 (1996)
34. Quinlan, J.R: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
35. Rabiner L.R. : A tutorial on hidden Markov models and selected applications in speech recognition. In: Proceedings of the IEEE 77, pp. 257 –286 (1989)
36. Michalski, R.S., Carbonell, J.G., Mitchell, T.M.: Machine Learning: An Artificial Intelligence Approach. Morgan Kaufmann, Tioga (1983)
37. Cheng, J., Greiner, R., Kelly, J., et al.: Learning bayesian networks from data: an information-theory based approach. Artif. Intell. **137**, 43–90 (2002)
38. Defays, D.: An efficient algorithm for a complete link method. Comput. J. **20**, 364–366 (1977)
39. Mitchell T.: Machine Learning. McGraw Hill, NY (1977)
40. Chittaro, L., Guida, G., Tasso, C., Toppano, E.: Functional and teleological knowledge in the multimodeling approach for reasoning about physical systems: a case study in diagnosis. IEEE Trans. Sys. Man Cybern. **23**, 1718–1751 (1993)
41. Le Goc, M.: SACHEM, a real-time intelligent diagnosis system based on the discrete event paradigm. Simulation **80**, 591–617 (2004)
42. Chittaro, L., Ranon, R.: Diagnosis of multiple faults with flow-based functional models: the functional diagnosis with efforts and flows approach. Reliab. Eng. Syst. Safety **64**, 137–150 (1999)
43. Zanni, C., Le Goc, M., Frydman, C.: A conceptual framework for the analysis, classification and choice of knowledge-based diagnosis systems. KES—Int. J. Knowl. Based Intell. Eng. Syst. **10**, 113–138 (2006)
44. Reiter, R.: A theory of diagnosis from first principles. Artif. Intell. **32**, 57–95 (1987)

45. Rosenberg, R.C., Karnopp, D.C.: Introduction to Physical System Dynamics. McGraw-Hill, NY (1983)
46. Chittaro, L., Ranon, R.: Augmenting the diagnostic power of flow-based approaches to functional reasoning. In: AAAI-96 Proceedings, pp. 1010–1015 (1996)
47. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**, 79–86 (1951)
48. Cheng, J., Bell, D., Liu, W.: Learning bayesian networks from data: an efficient approach based on information theory (1997)
49. Bouché, P., Le Goc, M., Coinu, J.: A global model of sequences of discrete event class occurrences. In: Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS 2008), pp. 173–180 (2008)
50. Fakhfakh I., Curt C., Le Goc M., Torrès L.: Diagnosis of the Hydraulic Dam Safety based on Multimodelling Approach. Actes du 18ème Congrès de Maîtrise des Risques et de Sûreté de Fonctionnement (2012)
51. Pomponio, L., Le Goc, M., Pascual, E., Anfosso, A.: Discovering models of human's behavior from sensor's data. In: Workshop Proceedings of the 7th International Conference on Intelligent Environments, pp. 17–28. IOS Press, Nottingham, 25–26 July 2011
52. Pomponio, L., Le Goc, M., Anfosso, A., Pascual, E.: Levels of abstraction for behavior modeling in the GerHome project. Int. J. E-Health Med. Commun. **3**, 12–28 (2012)
53. Pomponio, L., Le Goc, M., Pascual, E., Anfosso, A.: Resident's activity at different abstraction levels: proposition of a general theoretical framework. In: The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS'2011, pp. 540–545, Prague (2011)