UNIVERSITÉ AIX-MARSEILLE

Modeling and diagnosis of dynamic process from Timed Observations : Application to hydraulic dam

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ AIX-MARSEILLE

Faculté des Sciences

Discipline: Informatique

présentée et soutenue publiquement par

Ismail FAKHFAKH

??? 2014

Directeur de thèse: Marc LE GOC

Codirecteur de thèse: Lucile TORRES

Tuteur de thèse: Corinne CURT

École Doctorale en Mathématique et Informatique de Marseille
(ED 184)

Jury

M. Marc Le Goc	Professeur, Univ. Aix-Marseille, LSIS	Directeur de thèse
Mme. Lucile Torres	M.conf, Univ. Aix-Marseille, LSIS	Codirecteur de thèse
Mme. Corinne Curt	Ingénieur de recherche, IRSTEA	Tuteur de thèse

Année 2014

À mon fils Rayan, ma femme Nadia, mes Parents, Frère et Soeurs, mes beaux parents et mes beaux Frères et mes belles Soeurs ...

Acknowledgements

Je souhaite remercier en premier lieu mon directeur de thèse, M. Marc LE GOC, Professeur des Universités chez Aix-Mareseille Université et responsable du projet in4Be (Induction for Behavior Modeling) au laboratoire des sciences de l'information et des systèmes pour m'avoir accueilli au sein de son équipe. Je lui suis également reconnaissant pour le temps conséquent qu'il m'a accordé, ses qualités pédagogiques et scientifiques. J'ai beaucoup appris à ses côtés et je lui adresse ma gratitude pour tout cela.

J'adresse de chaleureux remerciements à mon co-encadrant de thèse, Mme Corinne CURT, Ingénieur de recherche à l'institut national de recherche en sciences et technologies pour l'environnement et l'agriculture, pour son attention de tout instant sur mes travaux, pour ses conseils avisés et son écoute qui ont été prépondérants pour la bonne réussite de cette thèse. Son énergie et sa confiance ont été des éléments moteurs pour moi.

Un grand merci à mon co-encadrant de thèse, Mme Lucile TORRES, Maitre de conférences du laboratoire des sciences de l'information et des systèmes, pour son implication dans le projet. Elle m'a beaucoup appris, j'ai apprécié son enthousiasme.

J'adresse de sincères remerciements à M. Laurent Peyras, directeur de l'unité de recherche OHAX à l'institut national de recherche en sciences et technologies pour l'environnement et l'agriculture pour m'avoir accueilli chaleureusement au sein de son équipe et pour son aide pendant la dernière année pour finaliser cette thèse.

Je voudrais remercier les rapporteurs de cette thèse ..., Professeur des Universités de ..., et M. ..., Professeur des Universités, ..., pour l'intérêt qu'ils ont porté à mon travail. J'associe à ces remerciements M. ..., pour avoir accepté d'examiner mon travail.

Je souhaite remercier spécialement ma petite famille mes deux amours : ma femme Nadia et mon fils Rayan pour leurs soutiens et leurs patiences tout au long de la thèse sans lesquels je n'en serais pas là aujourd'hui.

06 juin 2014, Marseille, France.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Hundreds of thousands of dams are now in use throughout the world and some of them have been operating for several centuries.

Dams represent important economic stakes due to the numerous roles they fulfill: storing water for irrigation, producing hydroelectricity, supplying water to towns and businesses, etc. Moreover, they contribute to the management of limited global water resources that are subject to poor distribution and considerable seasonal variations. Dams are heterogeneous structures featured by multiple behaviors that evolve through time because of their natural aging. This aging can be accelerated by environmental causes (climatic conditions, floods and earthquakes) or by internal causes (poor design or construction conditions, insufficient or inadequate maintenance...). These causes involve, during the life of the structure, the occurrence and the development of deterioration phenomena, more or less dependent and stemming from miscellaneous and complex sources. These degradation phenomena can jeopardize the dam safety possibly leading to the structure failure which can have dramatic consequences on the people and assets located downstream.

In order to increase the reliability and availability of the system, failures or faults resulting from the malfunctions of the system should be detected and repaired to remain the system functioning correctly and to keep its physical integrity. At the present, all over the world, the assessment of the performance and safety of dams, their diagnosis and proposals for corrective actions are carried out by expert engineers during dam reviews: numerous communications of the triennial congress organized by the International Committee on Large Dams present expert-based approaches for dam diagnosis [Cur13]. However, the acquisition of the necessary skills to carry out the diagnosis task manually can be long. As a consequence, it is relevant to develop support tools to help engineers with this task.

1.1 Diagnosis of Dam Safety

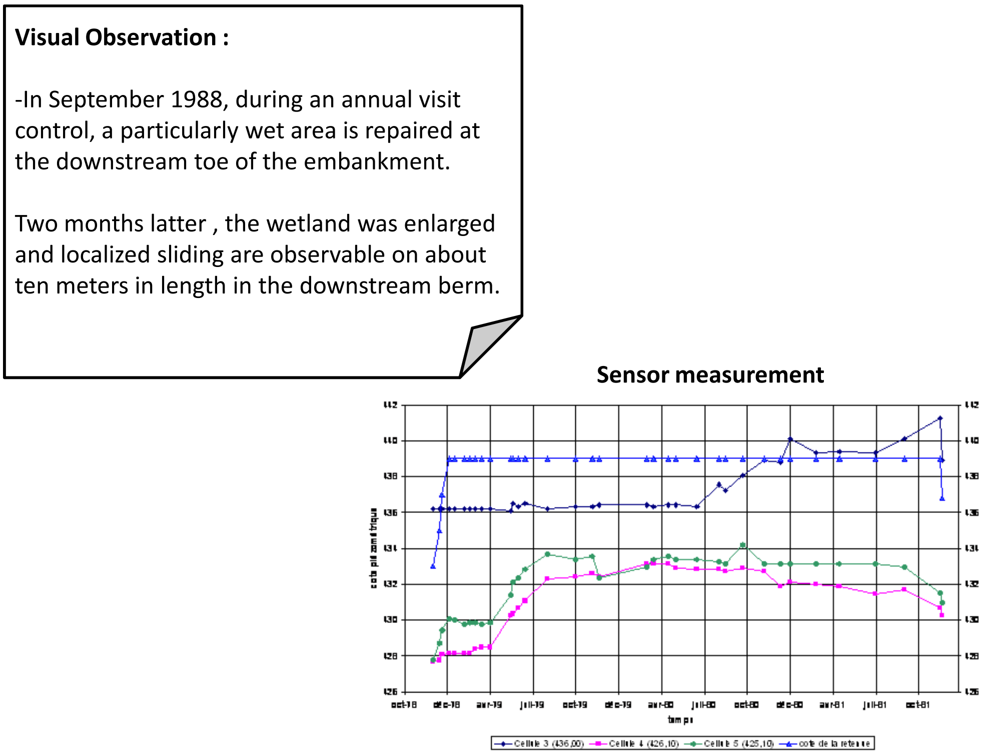
Regarding the former, performing the diagnosis of dam safety presents several difficulties linked to different features of the dam.

The first difficulty is about the multiplicity of the behaviors and the heterogeneity of the components that compose the hydraulic dam. Indeed, the hydraulic dam is composed by components which have a different functional role in the global system.

The second difficulty is about the temporal characteristics of the system which are the core of the solving problem method of dynamic systems like dams. Indeed, the evolution of the dam behavior is represented by a set of phenomena which present the possible states of the system over the time. The time can be either continuous (continuous signals as functions of time) or discrete (an interpretation by the human). The diagnosis approach method must allow taking into account this temporal characteristic.

The third difficulty is about the nature of the system knowledge. Indeed, dams are characterized by a several types of knowledge: knowledge coming from expert called visual observations (for example the detection of leakage on the downstream shoulder, detection of cracks on the facing, etc.), knowledge coming from monitoring devices called program observation like piezometry, flow change, etc., presented as mathematical models (hydraulic gradient, spillway capacity, etc.). It is important to note that visual observations are at the origin of the detection of most of dam anomalies [PR01].

The Knowledge Engineering (KE) discipline provides methods, techniques and tools which facilitate and improve the construction models based on the the expert knowledge. However, the model validation, if possible, only can be carried out with respect to that which the expert knows and makes explicit about the process, but not regarding the knowledge interpretation.



Visual Observation :

-In September 1988, during an annual visit control, a particularly wet area is repaired at the downstream toe of the embankment.

Two months latter , the wetland was enlarged and localized sliding are observable on about ten meters in length in the downstream berm.

Sensor measurement

Figure 1.1: Example of knowledge base available at the end of a dam review (extract)

Figure 1.1 presents an example of data used during a dam expertise, here it concerns the Cublize dam [PRB06]. Two types of measurements were performed on Cublize dam. The first ones concern visual observations describing on the one hand, the presence of a wet area on the downstream toe of the dam and its increase during time and on the other hand, the detection of a sliding of the downstream embankment. Both data indicate an abnormal presence of water in the work. The second type of measurement is instrumental ones: piezometry, flow and water level in the reservoir. Both types of measurement were necessary to carry out the diagnosis task.

This example reflects the usual way of processing performed by experts during reviews and show the necessity to develop an algorithm able to deal with these different kinds of measurement and that allow to combine the raw data process with the cognitive models used by the experts to assess the safety of dam.

The last difficulty is about the fault detection task. In fact, the presence of the faults must be known before they become serious. However, in some cases, the fault can not be known in advance. In the case of the dam, the fault is more likely some phenomenon we did not expect or an exogenous condition influencing the system behavior in an undesirable way. Besides, the phenomena can be interpreted differently from one expert to another. An abstraction representation of the diagnosis results is needed to deal with the different diagnosis objectives interpreted by the expert.

1.2 Model-Based Reasoning

The increasing complexity of engineered systems led the Model-Based Reasoning (MBR) community to focus its research on diagnosis tasks based on models with multiple levels of abstraction. Abstractions are useful for reducing the computational complexity of diagnosis reasoning, accounting for observations at qualitative levels, and handling systems whose available knowledge relating to components is heterogeneous.

Many approaches have exploited structural, behavior and functional models [CR04], [CGTT93] and [TMEPTS01]. The main claim of these approaches is certainly the requirement for the availability and use of a large variety of knowledge sources (models) linked to the physical system that is the object of the reasoning activity. In fact, it is widely recognized that no single model is adequate for a wide range of problem solving tasks.

However these approaches suffer from two principal drawbacks:

1. First, these approaches use the abstraction level to achieve different tasks (monitoring, diagnosis, prediction ...). Therefore, the abstraction level of the design task requires the definition of a lot of components, some of which might be meaningless in the diagnosis task. This leads to model a large number of components and hence a complexity to compute diagnosis system which is the objective of our works. Consequently, the main goal of the modeling task is to determinate the right level of abstraction in which the models have to be constructed to obtain a more efficient diagnosis since the number of diagnose increases exponentially with the number of components.

2. The second difficulty concerns the case of dynamic systems. Indeed, in these type of systems, the observation is timed, unlike in static systems where the observations are given at only one point of time. This is restrictive in several fields. The dynamic characteristics of the system poses many problems with the existing approaches to perform the diagnosis task.

Only very few approaches allows to reach these two requirements. Among these, the Timed Observation Theory [LG06] provides the mathematical and the methodological tools for the modeling, the supervision, the diagnosis and the prognosis of complex dynamic processes. In particular, the TOM4D methodology (Timed Observations Modeling For Diagnosis) based on a quadripartite model of a dynamic process that can be used for an efficient diagnosis task. One of the main advantages of the TOM4D methodology is its ability to combine raw process data with expert's knowledge to provide the abstract (or conceptual) model of a process an expert uses to formulate its diagnosis knowledge. TOM4D is then particularly well suited to provide an adequate model of dynamic and complex system to obtain an efficient diagnosis by implementing the models for diagnosis task.

To remedy these problems, dams deterioration and failure models and scenarios have been proposed in the literature. Nevertheless, these methods present some limits: they are qualitative, e.g., based on the FMEA (Failure Modes and Effects Analysis) approach [PRB06], or they only consider the future of the dam [FFS00], or they fail to take into account the whole set of available data, in particular visual data.

These limits have been encountered in many domains, notably in the computer science area where three main approaches of diagnosis are currently developed: the Logical Theory of Diagnosis, the Discrete Event System (D.E.S) and the Multi-Models approach.

- The logical Theory of Diagnosis, or diagnosis from first principles, was pioneered by Reiter [Rei87]. The basic idea is to predict the behavior of the system using behavioral and structural models of the system and its components and to compare it with observations of the actual behavior of the real system. Despite of the important contributions in the domain of temporal logic, there is still a difficulty to take into account the time of the observations in the diagnosis reasoning.

- In Discrete Event System community (D.E.S), the system is modeled as a finite state machine, where the states of the machine describe conditions of the components. The only dynamics represented in the FSM (Finite State Machine) is that the system can go from normal to faulty. The observations (sensor readings) are included as a mapping from the fault states to certain observable events (tests). To diagnose a fault corresponds to identify which state or set of states the system belongs to.

  The application of this approach on the diagnosis continuous time system implies a considerable number of discrete events. This leads to very large automate. The size of the diagnose can be exponential given the important number of states. Besides, these kinds of approaches say nothing about the visual observations which are the origin for many observations on systems like the dam. Finally, in the case of the dam diagnosis, it is very difficult to match the set of trajectories resulting from the diagnosis task with the set of phenomena interpreted by an expert. We believe then that the problem comes from the concepts used in D.E.S (event, state, and trajectory). Adding a sound cognitive and abstraction level to this approach is necessary to solve the problem. Nevertheless, establishing the meaning regarding the experts semantics of models obtained from data is complex.

- In Multi-Model based diagnosis, the system is modeled by means of three models: a structural model describing the components and their interconnections, a behavior model describing how components operate and interact with each other, and a functional model describing the different roles that the components can play in the physical process of which they are part. The large number of components of the resulting model leads to computing difficulties in the diagnosis task. The main problem is to determinate the right level of abstraction in which the models have to be built to obtain an efficient diagnosis. Most modeling approaches use the abstraction level of the available models, generally design models. However the abstraction level of the design task requires the definition of a lot of components, some of which might be meaningless in the diagnosis task. This results in a high computational cost since the number of possible diagnosis increases exponentially with the number of components. Define models that only consider components that are concerned with diagnosing allowing a more efficient diagnosis since the number of components is minimal.

With regard to the previously presented difficulties and context, the challenge lies in treating knowledge from different sources to define a modeling method for diagnosis and to compute the diagnosis objective for complex and dynamic system.

Therefore, we propose to develop a method able to deal with data of various types, to take into account the temporal characteristics of the system and applied for industrial system with different diagnosis objectives. The level of abstraction of the models is also a key feature: without a relevant level of abstraction, it is impossible to guarantee the global consistency of the diagnosis engine. An expert usually uses a set of models at a level of abstraction which is directly linked with the diagnosis task and not with the design task. The objective is thus to get an abstracted description of the system that is homogeneous with the expert abstraction level.

1.3 Contributions

The objective of our works is to propose a model-based diagnosis approach that takes into account the limitations presented in the community of Model-Based Diagnosis.

Only very few approaches allows to reach these requirements. Among these, the Timed Observation Theory [LG06] provides the mathematical and the methodological tools for the modeling, the supervision, the diagnosis and the prognosis of complex dynamic processes. This theory is important because it can be applied to any observed systems. The application of the TOT mathematical framework has given birth a modeling approach for diagnosis TOM4D (Timed Observations Modeling For Diagnosis). In particular, the TOM4D methodology allows to build a quadripartite model of a dynamic process that can be used for an efficient diagnosis task. One of the main advantages of the TOM4D methodology is its ability to combine raw process data with expert's knowledge to provide the abstract (or conceptual) model of a process an expert uses to formulate its diagnosis knowledge. TOM4D is then particularly well suited to provide an adequate model of a dam with the aim to ensure the compliance of its current state with the reliability and safety requirements, to diagnose the (past) causes of the deterioration of the reliability and the safety, and to forecast the reliability and safety evolutions at different time scales. Our first contribution is so concerned with the extension of the TOM4D methodology to networks of dynamic process.

Our second contribution is the design of diagnosis algorithm based on the TOM4D models. The proposed algorithm is based on the fact that observation models should be consistent with the available timed observations observation. Otherwise, an explanation for error needs to be provided. Therefore, we used the concept of observation class and path of observation class occurrences as the result of diagnosis algorithm by analogy with the notion of trajectory used in D.E.S. The flexibility of the notion of observation class path makes easier to connect the diagnosis results with different diagnosis purposes: identify the phenomena (in the case of the dams), the set of trajectories, the default components, the mode assignment ... (cf. Figure 1.2)

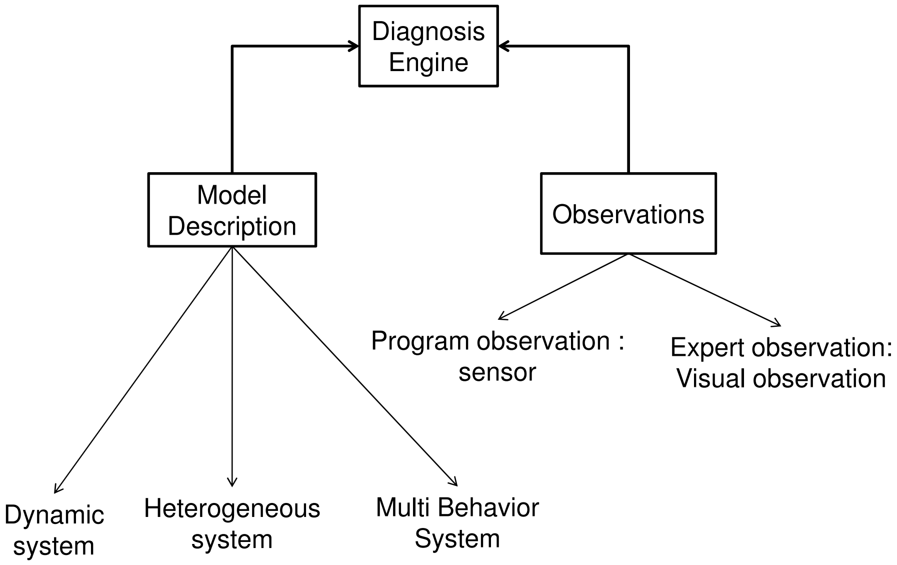


Figure 1.2: Diagnosis algorithm proposition

1.4 Document Structure

In the document, the problem statement of this thesis and our main contributions are described above in this introductory chapter.

The next chapter proposes a state of art concerning the three main diagnosis approaches: the Logical Theory of Diagnosis, the Discrete Event System (D.E.S) and the Multi-Models approach. To clarify the presentation and illustrate the particularities of the approaches, the example of an hydraulic system is used as a didactic example. This example will then be used all over this document in order to describe our contributions.

Chapter 3, is dedicated to an introduction of main concepts and theorems of the Timed Observation Theory which constitute the mathematical framework within our works are developed.

Chapter 4 presents the TOM4D methodology which is based on the idea that experts use implicit models to formulate their knowledge about a process and the way of diagnosing it. The systems concerned with this approach are typically characterized by a complex and heterogeneous structure that evolves over the time. The presentation of the TOM4D methodology is based on the didactic example described in the state of the art chapter.

Based on the models defined in TOM4D methodology, chapter 5 proposes an adequate diagnosis algorithm that allows identifying the faults occurring on a network of dynamic processes.

The objective of the diagnosis is to detect timely the faults based on observations of different natures (visual and numerical observations).

Chapter 6 is the real-world application chapter of this document. Both the TOM4D modeling approach and its corresponding diagnosis algorithm are applied to assess the hydraulic dam safety.This chapter shows the operational flavor of our contribution in a complex real-world problem.

Finally, Chapter 7 concludes this manuscript with a short synthesis of our contributions to position it in its global scientific context so that some perspectives of our works can be proposed.

CHAPTER 2

State of the Art

The diagnosis is the process of observing a system to detect any malfunctions from these observations and to identify their origin. Since the 60's, numerous different techniques have been proposed used to allow a computer to perform an automatic diagnosis. They come from research works in the fields of Artificial Intelligence (AI) and Automatic Control (AC) and they are used in various activity sectors, such as the medical sector or the industrial one. The problem that is concerned with this document relates to the diagnosis of a dam in service.

2.1 The Hydraulic Dam Diagnosis Problem

A dam is an art work established across the bed of a watercourse, for regulating the flow of water or for storing water. It is a construction that lives, works and deteriorates under the influence of forces, mainly pressure forces exerted by the water, to which it is subjected. Account held in the risk of dam failure and the extent of damage consequently on people, properties and environment, even if the probability of such a risk is extremely low, monitoring of dams to prevent accidents is a crucial task, highly regulated, which is the subject of numerous studies and research works.

The causes of rupture may be technical, when the failure is caused by a malfunction of the dewatering valves, by a faulty design, a construction defect or a defect in material, by the aging of the construction, or they may be natural, when the failure is the result of an earthquake, of an exceptional flood or of a landslide. They may even be human, resulting from a lack of prior studies, of execution control, of monitoring, of maintenance or resulting from an act of malice. The phenomenon of dam failure depends on the specific characteristics of the dam. Thus, the failure can be gradual, by retrogressive erosion, in the case of an embankment dam, or abrupt, by toppling or sliding of one or more blocks, in the case of a concrete dam. The risk is lower for concrete structures than for embankment ones. The accidents of dams identified by the International Commission on Large Dams (ICOLD) are classified by type of dam, by age and by cause of the accident. The most frequent causes of failure are [CIG95]:

- an excessive level of water upstream of the dam, due to improper spillway design, debris blockage of spillways, or settlement of the dam crest,
- foundation defects, including settlement and slope instability,
- the internal erosion caused by seepage,

- structural failures of the materials used in dam construction, resulting of the degradation of these materials.

The diagnosis of dams, with the risk analysis, is an essential task to ensure dam safety [PBRV04]. Its objective is to prevent the degradation of dams on the long term. A diagnosis task is firstly to detect deficiencies of the structure which may weaken it, such as cracks, irreversible deformations or leaks for example. And once the abnormality has been detected, it is interested in identifying the cause of this anomaly. In other words, it is interested in identifying the involved component whose knowledge is essential to define the actions to implement to simply monitor, to repair or to remove the dam. Diagnosis is made from many physical measurements or visual observations taken at different time on the structure: movements, piezometry, pore pressure, drainage flows or leaks, crack openings, etc. The various diagnosis approaches that have been implemented by civil engineers, have not necessarily been formalized with the concepts and in terms dedicated to system diagnosis as used in AI or AC. Nevertheless, there are approaches based on physical or functional dam modeling, and others that are based on statistical analysis of observed data or on expertise [PBR04, TBP09]. So it is possible to establish the same dichotomy as that between the methods of system diagnosis from AI and AC research, depending on the type of handled knowledge (Figure 2.1). The first category consists of methods based on the so called internal knowledge about the behavior of the dam considered as a set of interconnected components. In contrast, the second category consists of methods that exploit the knowledge that is said external knowledge this time, about the effects (symptoms) on the system allowing to establish correlations between faults and symptoms.

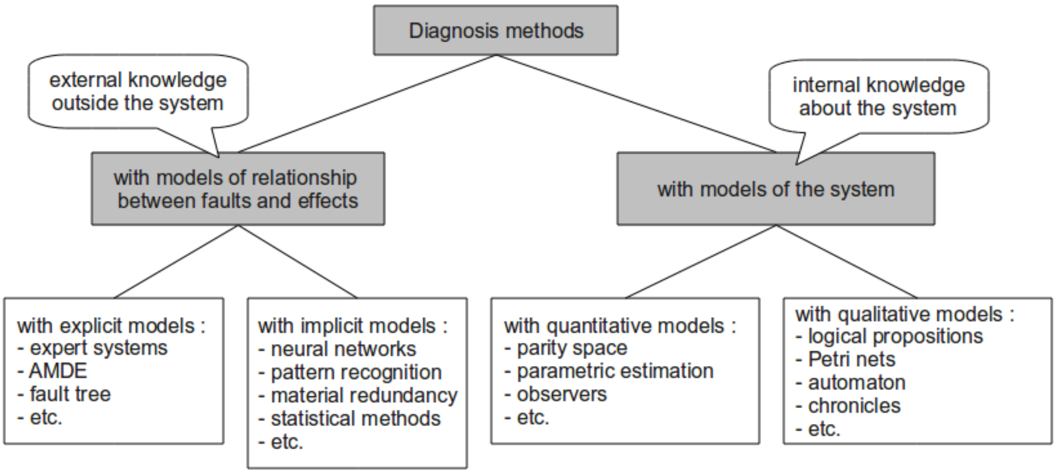

Diagnosis methods
external knowledge outside the system
internal knowledge about the system
with models of relationship between faults and effects
with models of the system
with explicit models : - expert systems - AMDE - fault tree - etc.
with implicit models : - neural networks - pattern recognition - material redundancy - statistical methods - etc.
with quantitative models : - parity space - parametric estimation - observers - etc.
with qualitative models : - logical propositions - Petri nets - automaton - chronicles - etc.

Figure 2.1: Classification of diagnostic methods based on the type of knowledge

The complexity of the diagnosis task and the need for performing it as soon as possible to prevent the consequences of faults, have motivated the attempts to automate this task. The methods developed in diagnosis tools vary according to the considered criteria: the used kind of knowledge, the way of structuring this knowledge, the involved scientific community and the application domain. The classification presented in Figure 2.1 represents only one of many possibilities of classification, the one which is established according to the type of knowledge.

Note that there is a variety of terms widely used to identify a fault in the field of diagnosis.

Then, we can find: default, failure, fault, malfunction, dysfunction, defect, anomaly, abnormality, etc. These terms correspond to very similar concepts but that are differentiated according to the implicated community and to the context specific to the application problem. While a fault is most often associated with an abnormal behavior of a physical component, a failure refers to the inability of the studied system to provide a required function under the specified operating conditions. For simplicity, we equivalently use all these terms to present the different diagnosis approaches.

This chapter starts with section 2.3 where a brief overview of the diagnosis methods that only use a model of faults (i.e. without using an explicit model of the system behavior). In section 2.4, we particularly focus on diagnosis based on analytical models of (normal or abnormal) behavior of the system, models that are the main diagnosis tool in the three diagnosis scientific communities: the FDI (as Fault Detection and Isolation) community that comes from the AC community, the DX (from the acronym of the International Workshop on Principles of Diagnosis) as a part of the AI community and the DES (Discrete Event System) community.

Whenever it is possible, the concepts used all along this document will be illustrated with the same didactic example provided by [CPR00]. The aim of this choice is both to facilitate the understanding of concepts with a unique example, before applying them on a real-world system which is, by nature, much more complex, and to allow the comparison of the different contributions proposed in this document. The following section (section 2.2) of the chapter describes this example.

2.2 A Didactic Example

This section introduces a simple but representative example of the hydraulic system of Figure 2.2 as it has been proposed by Lucas Console, Claudia Picardi, and Marina Ribaudo in [CPR00]. This example will be used as a guideline to present the various diagnostic techniques that are broached in this document.

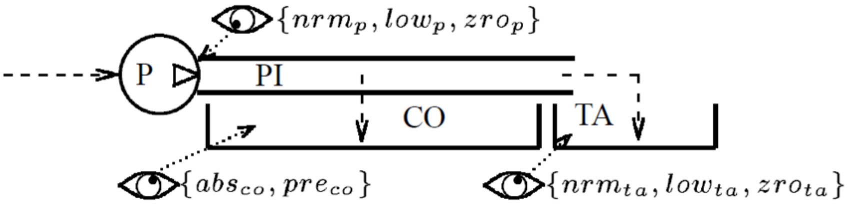


Figure 2.2: An hydraulic system

The hydraulic system *"is formed by a pump P which delivers water to a tank TA via a pipe PI ; another tank CO is used as a collector for water that may leak from the pipe. For the sake of simplicity we assume that the pump is always on and supplied of water.*

The pump P has three modes of behavior: ok (the pump produces a normal output flow), leaking (it produces a low output flow) and blocked (no output flow). The pipe PI can be ok (delivering to the tank the water it receives from the pump) or leaking (in this case we assume that it delivers to the tank a low output when receiving a normal or low input, and no output when receiving no input). The tanks TA and CO are always in mode ok, i.e., they simply receive *water.*

We assume that three sensors are available (see the eyes in Figure 2.2): $flow_p$ *measures the flow from the pump, which can be normal (*nrm_p*), low (*low_p*), or zero (*zro_p*);* $level_{ta}$ *measures the level of the water in TA, which can be normal (*nrm_{ta}*), low (*low_{ta}*), or zero (*zro_{ta}*);* $level_{co}$ *records the presence of water in CO, either present (*pre_{co}*) or absent (*abs_{co}*).”*

2.3 Diagnosis based on faults and their effects

Thanks to his ability to learn, to manage imprecise information and to decide in presence of a complex combination of information, the human expert demonstrates skills in fault detection and definition of a correct diagnosis, without needing any model on the system. The main idea of diagnosis based on a fault model is to build the diagnosis process by reproducing the human reasoning. Diagnosis based on faults is thus built from causal relationships between faults (causes) and effects on the system (symptoms). These causal relationships are the qualitative knowledge of the experts about the diagnosis process. Therefore, they are system-specific and empirical. They can be explicit, acquired from experts, expressed as rules in the case of an expert system, or represented by a fault tree. Otherwise, they are implicit, possibly automatically learned by machine learning techniques, and diagnosis is made by classification or pattern recognition. The diagnosis process only uses observations on the system to detect faults. The principle of diagnosis approaches based on a fault model is shown in Figure 2.3).

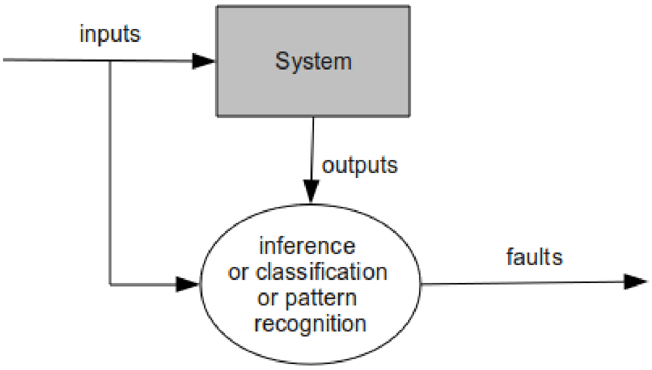


Figure 2.3: Principles of diagnosis based on the relationship between causes and effects

2.3.1 Diagnosis based on the causal relationships between faults and effects

These diagnosis approaches are based on a model of associative knowledge of the faults and of their effects. As they require the enumeration of all possible faults, they are closely linked to the diagnosis process and they mostly suffer from a lack of evolutivity and re-usability.

For instance, expert systems were developed int the mid 1960s and are mainly used as tools for decision support. They use rules that represent knowledge of the experts, specific to the problem resolution, of the form: *IF (P is true) AND (P* $\rightarrow$ *Q) THEN (Q is true)*. The first expert systems used for fault diagnosis are described in [SW89, Pau86]. Diagnosis on expert systems [Ang10] is made by deductive reasoning to infer new facts from certified facts and rules.

This reasoning is simple and provides explanations easily understandable by the users. Expert systems are poorly evolutive: any change in the problem or in the system undermines all the rules. Moreover, as the rules depend on the considered problem, they cannot be reused for another problem.

Failure Modes and Effects Analysis (FMEA) was developed in the 1950s to assure reliability and safety of a product or a process. It is an inductive approach: the faults caused by an effect observed on an entity of the system are determined from this effect. The method identifies the opportunities for failure, the causes of those failures and the effects on system performance. This aim is both to identify the points of failure and to classify failures by impact. Failures are prioritized according to the severity of their consequences, their frequency, and the ease with which they can be detected. FMEA has yet been applied on diagnosis of dams [PBRV04].

The fault tree [LY77, PL76, TP79], is a diagnosis method based on a tree structure of the possible combination of events that result in the production of an undesirable event. The undesirable event (the fault) constitutes the top of the tree, the immediate causes that produce this event are introduced into the tree by using the logical operators *and* and *or*. And so on, the tree is incrementally built until the basic events are added as leaves of the tree. The fault tree is suited for top-down analysis, aimed at analyzing the effects of initiating faults on a system, and it is not good for finding all possible faults. In contrast, FMEA is a bottom-up analysis method aimed at analyzing the effects of faults of a single component on a subsystem.

2.3.2 Diagnosis by classification or pattern recognition

Some works have been conducted to reduce the dependence of the diagnosis task to the expertise of a human operator. It is practically to acquire the diagnosis principles with no need for the expert. These works proceed by supervised or not supervised machine learning. Such approaches assume that the available knowledge on the system is limited to its past and present observation. And the fault model useful for the diagnosis process is built by classification, from the collection of data during system operating. These approaches are highly dependent on the volume of data ad are often limited to fault detection.

The supervised machine learning methods such as neural networks [Bis94, NP90] allow to discover a model of causal relations from a history of the observations where the symptoms have previously been diagnosed and labeled by the corresponding causes. Then the model is used to find the causes of dysfunction in new cases of operation. The causes are determined through the model, by probabilistic or fuzzy classification. The major drawback of this type of method lies with the learning time [Dub01].

When faults have not previously been diagnosed, it is still possible to make a categorization of faults by means of statistical (unsupervised) classification. The only available knowledge about the process is a set of observations and fault discrimination depends on a large volume of observations. These statistical methods assume that the values provided by the sensors have certain statistical characteristics on which thresholds can be calculated for example, from the study of the evolution of the mean or the variance of a variable, the exceeding of a threshold indicating the presence of an anomaly.

The automatic acquisition of correlation rules requires less expertise than developing an expert system. By cons, without this expert knowledge, it is very difficult to assign a real explanation (a fault diagnosis) to an observed and recognized situation (symptoms). By using the observations of the system rather than the structure and the behavior of the system, systems of automatic acquisition of diagnosis models produce a weakly intuitive and explanatory diagnosis. Therefore, in order to make the acquisition of diagnosis principles automatic without the intervention of the expert, it becomes essential to rely on the knowledge of a lower level, these that describes the system structure and behavior. And the words 'system model' used to describe these internal knowledge are opposed to 'fault model' which represent the external knowledge in terms of associations between faults and effects.

2.4 Model-based diagnosis

Model-based diagnosis (MBD) refers to methods for which diagnosis is built from a model of the system (and not a model of faults) [MK92]. MBD was born in the 1980s with the work of Raymond Reiter [Rei87] and Johan de Kleer [DKW87]. This kind of diagnosis is implemented by an algorithm that generates diagnosis hypotheses using both the observations measured on the real system and the ones measured on a model of this sytem that describes its normal or abnormal behavior. The system is generally represented in terms of components, interconnected by their inputs and outputs. Unlike diagnosis methods exclusively based on faults, MBD requires no prior knowledge of faults, it is based on a comparison between the actually observed behavior and the expected behavior of the system ([CD99], discrepancy detection problem in [DH92]). The common principles of MBD are illustrated in Figure 2.4.

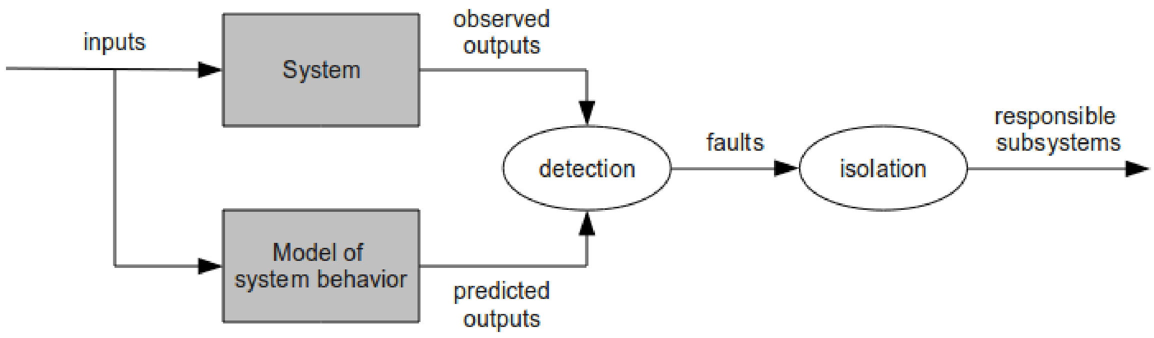


Figure 2.4: MBD principles

Detecting the presence of defects is the essential step, prior to the proper diagnosis of these defects, that is to say prior to their localization and to the identification of their causes. Defect detection determines the normality or abnormality of the system functioning, by comparing observations of the actual behavior of the system, obtained via sensors, with predictions of its expected behavior, calculated with the model of good or bad behavior. When the model only describes the normal behavior of the system, any deviation or inconsistency between the observed behavior and the expected behavior is a manifestation of a failure. Otherwise, the detection is done by recognizing the observations as faults that are recorded in the model of bad behavior. The complementary phase of detection involves refining the location of the faults until the subsystems responsible of the faults are found.

Describing the system only by its normal behavior, by dispensing with any description of its failure modes, is a part of a rather favorable approach to the design of the system, together with its monitoring process. The difficulty is to correctly and fully describe the functioning of the system. The description of the abnormal behavior of the system depends, in turn, on the ability to predict all the faults of the system, and the difficulty lies in the completeness of the description. An error in the model of normal or abnormal behavior inevitably results in either the undesired detection of faults that are not, or in the inability to detect certain faults. The validity of the model, that is to say its completeness and the absence of errors, is among the first assumptions of MBD.

A multitude of MBD methods have been developed, which have resulted in numerous comparative studies [CDD+00b, CDL+04, Nyb01]. These methods can be differentiated according to the type of models they use:

- Methods of the DX approach from AI research, based on qualitative models expressed as logical propositions;
- Methods for continuous systems of the FDI approach from AC research, using quantitative models, often expressed in terms of differential algebraic equations;
- Methods for DES, which also employ qualitative models, based on purely discrete representation of the system operation, by exploiting the discrete nature of the variables (e.g. open and closed values of a valve) or by abstracting continuous dynamics in discrete, qualitative and finite states.

2.4.1 Logical approach of diagnosis

Diagnosis approach in the logical approach defined by the DX community is to determine, given a description of a system and behavioral observations of this system conflicting with the expected behavior, the system components which restore consistency with the observed behavior when they are considered faulty [Rei87]. In logical theories as they were introduced by Reiter [Rei87] and De Kleer [DKW87], the behavior of the system components and the observations are described in first-order logic. Many studies have subsequently improved these theories by incorporating more expressive languages [DKMR92]. In this section, the logical framework of MBD as originally defined by Raymond Reiter is recalled without addressing the variants of later approaches.

2.4.1.1 Diagnosis problem

The system model is designed independently of diagnosis purposes. It expresses how the system functions when its normally behaves. Since the diagnosis goal is to find the part (component) of the system that is responsible for a given abnormal behavior, the model must be structured into components, so that the responsibility for some of the behavior of the system can be assigned to each component.

Definition 2.1 *The description of the system is defined by a triplet* $SD = (\ COMPTYPES\ ,\ CONNS\ ,\ TYPEBEHAVIORS\ )$ *where, if* $COMPS$ *means all names of system components:*

- $COMPTYPES = \{type_i(c)/c \in COMPS\}$ *is the set of system components with their types, where the predicate* $type_i(c)$ *means ”component c is of type i”;*

- $CONNS = \{in_i(c) = out_j(d)/c \in COMPS, d \in COMPS\}$ *is the set of relations between the inputs and the outputs of the components, where the predicate* $in_i(c) = out_j(d)$ *means "an input of component c is an output of component d" et* in_i *(respectively* out_j*) is the function which associates one of the inputs of a component (respectively one of its outputs) to the component;*

- $TYPEBEHAVIORS = \{type_i(x) \wedge ok(x) \rightarrow \Phi(x)/x \in COMPS\}$ *is the set of normal behaviors of the component types, where* $type_i(x) \wedge ok(x) \rightarrow \Phi(x)$ *means "for any component x, if x is of* $type_i$ *and is assumed to normaly behave, then it behaves as described by the first order logical formula* $\Phi(x)$*".*

Predicates $ok(x)$ et $\neg ok(x)$ imply that component x normaly behaves and it abnormaly behaves, respectively.

Example 2.1 *Let us consider the hydraulic system introduced in section 2.2. We can distinguish four types of components (*$COMPTYPES_1$*) making up this system and four components (*$COMPS_1$*, one for each type) connected together according to* $CONNS_1$*:*

$$\begin{aligned} COMPS_1 &= \{c_1, c_2, c_3, c_4\} \\ COMPTYPES_1 &= \{pump(c_1), pipe(c_2), tankTA(c_3), tankCO(c_4)\} \\ CONNS_1 &= \{out_1(c_1) = in_1(c_2), out_1(c_2) = in_1(c_3), out_2(c_2) = in_1(c_4)\} \end{aligned}$$

The normal behavior of each component is shown by :

$TYPEBEHAVIORS_1 = \{$

$$\begin{aligned} pump(x) \wedge ok(x) &\rightarrow out_1(x) = nrm_p \ , \\ pipe(x) \wedge ok(x) &\rightarrow (in_1(x) = nrm_p \vee in_1(x) = low_p \vee in_1(x) = zro_p) \\ &\quad \wedge out_1(x) = in_1(x) \wedge out_2(x) = zro_p \ , \\ tankTA(x) \wedge ok(x) &\rightarrow (in_1(x) = nrm_p \vee in_1(x) = low_p \vee in_1(x) = zro_p) \\ &\quad \wedge (in_1(x) = nrm_p \rightarrow out_1(x) = nrm_{ta}) \\ &\quad \wedge (in_1(x) = low_p \rightarrow out_1(x) = low_{ta}) \\ &\quad \wedge (in_1(x) = zro_p \rightarrow out_1(x) = zro_{ta}) \ , \\ tankCO(x) \wedge ok(x) &\rightarrow (in_1(x) = low_p \vee in_1(x) = zro_p) \\ &\quad \wedge (in_1(x) = low_p \rightarrow out_1(x) = pre_{co}) \\ &\quad \wedge (in_1(x) = zro_p \rightarrow out_1(x) = abs_{co}) \end{aligned}$$

$\}$

$MS1 = (COMPTYPES_1, CONNS_1, TYPEBEHAVIORS_1)$ *provides the complete model of the hydraulic system, ready for diagnosis process.*

A diagnosis is necessary as soon as there is a deviation between the observed behavior of the system and its expected behavior as described by the model [DH92]. Since system behavior is described by a set of logical formulas, this difference is revealed when the logical formulas representing observations on the system are inconsistent with the assumption that all components are normally functioning.

Definition 2.2 *A diagnosis problem is defined by a triplet* $(SD, COMPS, OBS)$ *where SD is a description of the system, COMPS is a set of component names designating the components of the system, and OBS is a set of first order logical formulas expressing observations on the system, such as the set of formulas* $SD \cup \{ok(c)/c \in COMPS\} \cup OBS$ *is inconsistent.*

Example 2.2 *Let us consider the triplet $(SD_1, COMPS_1, OBS_1)$ with SD_1 et $COMPS_1$ corresponding to the sets obtained for the system hydraulic in Example 2.1, and $OBS_1 = \{out_1(c_1) = nrm_p, out_1(c_3) = low_{ta}, out_1(c_4) = pre_{co}\}$. With the hypthesis that all the components are correct and given the observation on the value $out_1(c_1)$ (that is to say $flow_p$), it is possible to obtain the expected values of the system, $out_1(c_3)$ (actually $level_{ta}$) et $out_1(c_4)$ (actually $level_{co}$), by simple derivation :*

- $out_1(c_1) = nrm_p$, $out_1(c_1) = in_1(c_2)$

  $\vdash in_1(c_2) = nrm_p$

- $pipe(c_2) \wedge ok(c_2), pipe(x) \wedge ok(x) \rightarrow (in_1(x) = nrm_p \vee in_1(x) = low_p \vee in_1(x) = zro_p) \wedge out_1(x) = in_1(x) \wedge out_2(x) = zro_p$

  $\vdash (in_1(c_2) = nrm_p \vee in_1(c_2) = low_p \vee in_1(c_2) = zro_p) \wedge out_1(c_2) = in_1(c_2) \wedge out_2(c_2) = zro_p$

- $in_1(c_2) = nrm_p, (in_1(c_2) = nrm_p \vee in_1(c_2) = low_p \vee in_1(c_2) = zro_p) \wedge out_1(c_2) = in_1(c_2) \wedge out_2(c_2) = zro_p$

  $\vdash out_1(c_2) = nrm_p \wedge out_2(c_2) = zro_p$

- $out_1(c_2) = in_1(c_3), out_1(c_2) = nrm_p$

  $\vdash in_1(c_3) = nrm_p$

- $tankTA(c_3) \wedge ok(c_3), tankTA(x) \wedge ok(x) \rightarrow (in_1(x) = nrm_p \vee in_1(x) = low_p \vee in_1(x) = zro_p) \wedge (in_1(x) = nrm_p \rightarrow out_1(x) = nrm_{ta}) \wedge (in_1(x) = low_p \rightarrow out_1(x) = low_{ta}) \wedge (in_1(x) = zro_p \rightarrow out_1(x) = zro_{ta})$

  $\vdash (in_1(c_3) = nrm_p \vee in_1(c_3) = low_p \vee in_1(c_3) = zro_p) \wedge (in_1(c_3) = nrm_p \rightarrow out_1(c_3) = nrm_{ta}) \wedge (in_1(c_3) = low_p \rightarrow out_1(c_3) = low_{ta}) \wedge (in_1(c_3) = zro_p \rightarrow out_1(c_3) = zro_{ta})$

- $in_1(c_3) = nrm_p, (in_1(c_3) = nrm_p \vee in_1(c_3) = low_p \vee in_1(c_3) = zro_p) \wedge (in_1(c_3) = nrm_p \rightarrow out_1(c_3) = nrm_{ta}) \wedge (in_1(c_3) = low_p \rightarrow out_1(c_3) = low_{ta}) \wedge (in_1(c_3) = zro_p \rightarrow out_1(c_3) = zro_{ta})$

  $\vdash out_1(c_3) = nrm_{ta}$

- $out_2(c_2) = in_1(c_4), out_2(c_2) = zro_p$

  $\vdash in_1(c_4) = zro_p$

- $tankCO(c_4) \wedge ok(c_4), tankCO(x) \wedge ok(x) \rightarrow (in_1(x) = low_p \vee in_1(x) = zro_p) \wedge (in_1(x) = low_p \rightarrow out_1(x) = pre_{co}) \wedge (in_1(x) = zro_p \rightarrow out_1(x) = abs_{co})$

  $\vdash (in_1(c_4) = low_p \vee in_1(c_4) = zro_p) \wedge (in_1(c_4) = low_p \rightarrow out_1(c_4) = pre_{co}) \wedge (in_1(c_4) = zro_p \rightarrow out_1(c_4) = abs_{co})$

- $in_1(c_4) = zro_p, (in_1(c_4) = low_p \vee in_1(c_4) = zro_p) \wedge (in_1(c_4) = low_p \rightarrow out_1(c_4) = pre_{co}) \wedge (in_1(c_4) = zro_p \rightarrow out_1(c_4) = abs_{co})$

  $\vdash out_1(c_4) = abs_{co}$

Thus, $SD_1 \cup \{ok(c)/c \in COMPS_1\} \cup \{out_1(c_1) = nrm_p\} \vdash out_1(c_3) = nrm_{ta}$
$SD_1 \cup \{ok(c)/c \in COMPS_1\} \cup \{out_1(c_1) = nrm_p\} \vdash out_1(c_4) = abs_{co}$

The value of the output of c_3, as predicted by the model, is nrm_{ta}, whereas its observed value is given in OBS_1 and is low_{ta}. And the value of the output of c_4, predicted by the model, is abs_{co} , whereas its observed value, given in OBS_1, is pre_{co}. Therefore, $SD_1 \cup \{ok(c)/c \in COMPS_1\} \cup OBS_1$ is inconsistent and $(SD_1, COMPS_1, OBS_1)$ is a diagnosis problem.

The diagnosis is based on the notion of conflict [Rei87, DKW87] between the observations of the system and the predictions made by the model. A minimal set of components that restores the consistency with observation once you stop to consider them correct, is precisely called a diagnosis based on consistency. This is the minimal set of components such that, if we replace $ok(c)$ by $\neg ok(c)$ for each component c in the diagnosis problem, the set of formulas describing the system and the observations, becomes consistent.

Definition 2.3 *A set of component names $\Delta \subseteq COMPS$ is a diagnosis for the diagnosis problem $(SD, COMPS, OBS)$, if Δ is a minimal set such that $SD \cup \{ok(c)/c \in COMPS - \Delta\} \cup \{\neg ok(c)/c \in \Delta\} \cup OBS$ is consistent.*

Example 2.3 *We want to make a diagnosis for the diagnosis problem of Example 2.2. Assuming that the component c_3 is defective, in other words deleting the hypothesis ok for this component, amounts somehow to remove the description of its behavior of the desciption of the system. Its output is then unknown and can take any value. In particular, the model no longer implies that $out_1(c_3) = nrm_{ta}$ and the consistency with the observation $out_1(c_3) = low_{ta}$ is restored. In the same way, if we suppose that component c_4 is faulty, the consistency is restored with the observation $out_1(c_4) = pre_{co}$. Therefore, $\{c_3, c_4\}$ is a diagnosis for our problem.*

c_2 being faulty, its outputs become unknown. Because they are used to determine the outputs of c_3 and c_4, the latter are also unknown. The outputs of c_3 and c_4 are indeterminate and they accept any values. The consistency with OBS_1 est recovered and $\{c_2\}$ is a diagnosis for our problem too.

Finally, the set of the diagnoses for this problem is : $\{\{c_1\}, \{c_2\}, \{c_3, c_4\}\}$.

The diagnosis algorithm is the way to calculate the diagnoses of a problem diagnosis, which meet the Definition 2.3 stated above.

2.4.1.2 Diagnosis algorithm

A diagnosis in the logical approach of diagnosis is obtained by using a model of the system where only the normal behavior of the components is detailed. It is defined in Definition 2.3, as a minimal set of components such as a superset of this diagnosis is not necessarily a diagnosis. This definition of diagnosis cannot be directly usable at the algorithmic level. It is interesting to take into account the following property of a diagnosis to focus the diagnosis algorithm on finding maximal subsets of non-defective components, rather than looking for minimal subsets of defective components.

Theorem 2.1 *A set $\Delta \subseteq COMPS$ is a diagnosis for the diagnosis problem $(SD,\ COMPS,\ OBS)$, iff Δ is a minimal set (in other words $COMPS - \Delta$ is a maximal set) such that $SD \cup \{ok(c)/c \in COMPS - \Delta\} \cup OBS$ is consistent.*

The diagnosis is then computed in breadth-first traversing the lattice formed by the set of subsets of $COMPS$ ordered by inclusion. This lattice is the search space of the diagnosis algorithm [DKW87].

Example 2.4 *The lattice for the set $COMPS_1$ of the diagnosis problem of Example 2.2 is graphically represented by the Hasse diagram in Figure 2.5.*

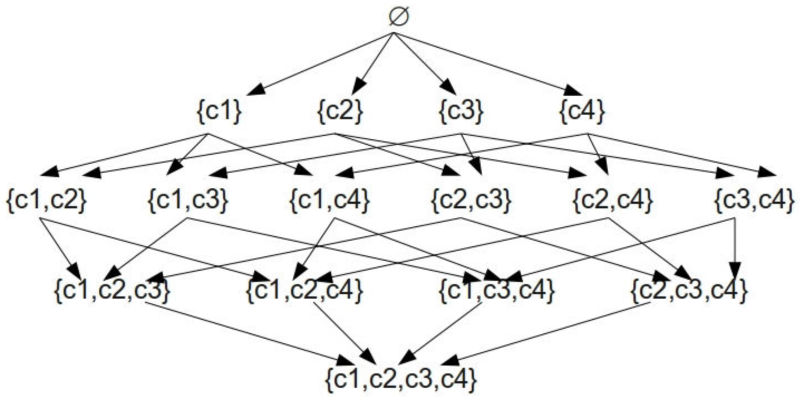


Figure 2.5: Search space of subsets of components implicated in the diagnosis of the hydraulic system

And the diagnosis algorithm is as follows.

Algorithm 2.1 *for each subset X of the breadth-first, left to right traversal of the lattice*

```
starting from the root (that is to say from the empty subset $\emptyset$)
    if $SD \cup \{ok(c)/c \in COMPS - X\} \cup OBS$ is consistent
    then
        $X$ is a diagnosis
        eliminate its descendants (as supersets of $X$)
    else (that is to say if $COMPS - X$ is a conflict as defined by Reiter [Rei87])
        eliminate the nodes $Y$ such that $Y \cap COMPS - X = \emptyset$ ($Y$ cannot be a diagnosis)
    endif
endfor
```

This algorithm has been optimized to avoid browsing all the nodes of the lattice, by using the concept of conflict that defines a conflict as a set of components which cannot all normally behave, and by using the property of a diagnosis, which says that the intersection of a diagnosis with any conflict is not empty.

Example 2.5 *Let us apply this algorithm to the diagnosis problem of Example 2.2.*

- *For $X = \emptyset$, $COMPS_1 - X = \{c_1, c_2, c_3, c_4\}$ is a conflict, therefore $\emptyset$ is not a diagnosis.*
- *For $X = \{c_1\}$, since $COMPS_1 - X = \{c_2, c_3, c_4\}$ is not a conflict, $\{c_1\}$ is a diagnosis and its descendants in the lattice will not be browsed, the concerned descendants are : $\{c_1, c_2\}, \{c_1, c_3\}, \{c_1, c_4\}, \{c_1, c_2, c_3\}, \{c_1, c_2, c_4\}, \{c_1, c_3, c_4\}, \{c_1, c_2, c_3, c_4\}$.*

- *For* $X = \{c_2\}$*,* $COMPS_1 - X = \{c_1, c_3, c_4\}$ *is not a conflict,* c_2 *is a diagnostic and the remaining descendants in the lattice are removed :* $\{c_2, c_4\}, \{c_2, c_3, c_4\}$*.*

- *For* $X = \{c_3\}$*,* $COMPS_1 - X = \{c_1, c_2, c_4\}$ *is a conflict, the following node is deleted :* $\{c_4\}$*.*

- *For* $X = \{c_3, c_4\}$*, since* $COMPS_1 - X = \{c_1, c_2\}$ *is not a conflict,* $\{c_3, c_4\}$ *is a diagnosis. We get the three diagnoses we set out in Example 3 :* $\{c_1\}, \{c_2\}, \{c_3, c_4\}$*.*

2.4.1.3 Diagnosis with dysfunctioning modes

Although the original idea of the logical approach of diagnosis is to pass knowledge about defects and their effects, such a knowledge, when available, is essential to identify the defects after having localized them. Extensions of the logical theory of diagnosis [CT90, CT91, SD89, DKW89] were then proposed so that the system model describes different types of malfunction, in addition to the normal behavior. Several behavioral modes can be now defined for each component. Indeed, instead of having only two behavioral modes per component (normal and abnormal) such that only the normal one is modeled, several faulty modes are added to the normal mode [?]. For this, the notion of behavior of a component has to be changed. For each component type t, we introduce a predicate associated with each behavior mode : ok for normal behavior of the component type t, and as many predicates as defective modes of the type t. An unknown mode may be added to account the impossibility of describing all possible faults, devoid of any behavior model, and supposed to bring together all unlisted faulty behaviors.

Example 2.6 *If we still look at the hydraulic system, the pump has now three listed operation modes, ok, leaking and blocked:*

$$\begin{array}{lcl} pump(x) \wedge ok(x) & \rightarrow & out_1(x) = nrm_p \\ pump(x) \wedge leaking(x) & \rightarrow & out_1(x) = low_p \\ pump(x) \wedge locked(x) & \rightarrow & out_1(x) = zro_p \end{array}$$

And the behavior of any component type is provided by :

$$
\begin{array}{rcl}
\multicolumn{3}{l}{TYPEBEHAVIORS_2 = \{} \\
pump(x) \wedge ok(x) & \rightarrow & out_1(x) = nrm_p, \\
pump(x) \wedge leaking(x) & \rightarrow & out_1(x) = nrm_p, \\
pump(x) \wedge blocked(x) & \rightarrow & out_1(x) = zro_p, \\
pipe(x) \wedge ok(x) & \rightarrow & (in_1(x) = nrm_p \vee in_1(x) = low_p \vee in_1(x) = zro_p) \\
 & & \wedge out_1(x) = in_1(x) \wedge out_2(x) = zro_p, \\
pipe(x) \wedge leaking(x) & \rightarrow & (in_1(x) = nrm_p \vee in_1(x) = low_p \vee in_1(x) = zro_p) \\
 & & \wedge((in_1(x) = nrmp \vee in_1(x) = low_p) \rightarrow \\
 & & out_1(x) = lowp \wedge out_2(x) = zro_p) \\
 & & \wedge(in_1(x) = zro_p \rightarrow out_1(x) = zro_p \wedge out_2(x) = zro_p), \\
tankTA(x) \wedge ok(x) & \rightarrow & (in_1(x) = nrm_p \vee in_1(x) = low_p \vee in_1(x) = zro_p) \\
 & & \wedge(in_1(x) = nrm_p \rightarrow out_1(x) = nrm_{ta}) \\
 & & \wedge(in_1(x) = low_p \rightarrow out_1(x) = low_{ta}) \\
 & & \wedge(in_1(x) = zro_p \rightarrow out_1(x) = zro_{ta}), \\
tankCO(x) \wedge ok(x) & \rightarrow & (in_1(x) = low_p \vee in_1(x) = zro_p) \\
 & & \wedge(in_1(x) = low_p \rightarrow out_1(x) = pre_{co}) \\
 & & \wedge(in_1(x) = zro_p \rightarrow out_1(x) = abs_{co}) \\
 & \} &
\end{array}
$$

Diagnoses are no longer sets of components that are supposed to be faulty to restore the consistency [DKMR92]. They have become assignations of behavior modes to components. And a diagnosis consists in finding a set of causes (components with a particular operation mode) which implies the symptom defined by the observations. However, some observations, those on the system inputs, cannot be deduced from the model. They must therefore be added to the model to enable deductions on outputs, they define the execution context of the model. As a result, they are separated from observations that relate to outputs and actually represent symptoms.

Example 2.7 *The diagnosis problem of Example 2.6 is converted to abductive diagnosis by setting the following set of observations :*

- *the context (observations on inputs) :* $CNTXT = \{out_1(c_1) = nrm_p\}$
- *the sysmptoms (observations on outputs) :* $SMPTM = \{out_1(c_3) = low_{ta}, out_1(c_4) = pre_{co}\}$

And the set of abductive diagnoses for this problem is : $\{\{leaking(c_2)\}\}$

With respect to the DX approach, we have seen that it relies upon a logically based theory for diagnosis of static systems. From a logical point of view, fault detection is performed through a consistency check between actual system behavior and observations, organized around the conflict idea [Rei87, DKW87]. In this approach, fault isolation is automatically derived from the phase of conflict detection. As the model of the system (the so called system description) is based on components, the aim of diagnosis is to find the faulty components of the system. However, there are two ways to complete fault isolation within the DX community : a consistency-based technique or an abduction-based one. While consistency-based diagnosis tries to reject the behavior modes of the components which are not consistent with current observations, abduction-based diagnosis tries to explain current observations by assignating a consistent behavioral mode to the components.

We will see that, although the basic principles of diagnosis are the same, the FDI community has developed its own concepts, tools and techniques, guided by a different context of modeling, quantitative rather than qualitative.

2.4.2 Diagnosis approach by analytical redundancy

The principle of redundancy in the FDI diagnosis approach [Dub01] is to make available several means to acquire a same information to diagnose a fault. The analytical redundancy, as opposed to material redundancy (of sensors for example), provides the redundancy to detect faults from the available observations through the system model. Diagnosis, in the approach proposed by the FDI community, is thus to generate indicators of faults, called residues, measuring deviations from the normal behavior of the system determined by the system model. A residue is a numerical value that expresses incompatibility or inconsistency between measurements made on the observed variables of the system, and calculations on the system model. The main quality of a residue is that it is sensitive of faults and insensitive to disturbances (measurement noise for example).

Diagnosis approaches based on residues are primarily dedicated to the detection phase which comprises generating the most telling symptoms of the current state of the system. The analysis of the obtained residues allows at least for determining whether or not there is a fault (detection), on which component of the system the default has occurred (localization), and possibly what is the nature of the fault (identification). Most of the time, residue analysis is performed using a threshold, fixed by a learning technique, beyond witch the residue is declared abnormal. Identifying the thresholds can be challenging : a too large value will prevent the detection of some faults, while a too small value will cause the inadvertent detection of nonexistent faults. There are alternative methods to direct comparison of a residue (or of the mean of the residues) to a threshold, dedicated to the evaluation of residues : parity space methods [CW84, GS90, Ger97], observers methods [Bea71, Jon73, AF97, Fra96], parameter estimation methods , etc. However, most methods of detection boil ultimately down to thresholding. In some cases, the assessment of the presence of a fault is not boolean, and consists in assigning a level of belief to dysfunction hypotheses, diagnosis must then uses techniques from fuzzy logic [Zad65].

2.4.2.1 Diagnosis problem

Most of the work of the FDI community does not explicitly use the concept of component to describe the system. The model describes the system as a whole using variables and outlines existing constraints between these variables. The formalization of the constraints on the variables depends on the type of knowledge : analytical knowledge, production rules, numeric tables, etc. It is typically provided in the form of differential algebraic equations. Among the variables describing the system, it is usual to distinguish between (observable) known variables, consisting of measured variables and reference variables, and (unobservable) unknown variables.

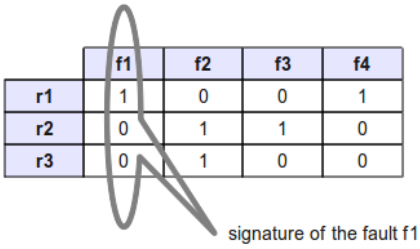
Definition 2.4 *The system model is defined by a pair $SM = (BM, OM)$ made up by a behavioral model (BM) representing constraints on variables and an observation model (OM) which lists the observable variables of the system.*

Definition 2.5 *A diagnosis problem is defined by the model of the system SM, a set of observations OBS assigning values to observed variables and a set of faults F where a fault corresponds to a faulty component or a set of faulty components [CDD+00a].*

We will ensure that the observed behavior of the system is normal or abnormal, by comparing it with that obtained using the system model. The behavior is considered abnormal when the observations and the data from the model are inconsistent. For a given diagnosis problem, the consistency of observations with respect to the model is established by checking that the observed behavior satisfies specific constraints, built from the model and called analytical redundancy relations. An analytical redundancy relation [FD00, Ger98, IB97, Pat97] is a constraint that contains only observed variables, so it can always be evaluated from the set of observations OBS. It is obtained from the model by removing the unknown variables. This is a function of known variables that satisfies the behavior of the model when its result (residue) exceeds a certain tolerance threshold ([BSL01, KN02, EMG98]). Analytical redundancy relations must be calculated with the observed values, and achieving a residue which exceeds a threshold is expected to clearly set out the presence of a defect. The notions of residue and threshold are provided for reducing the value ranges of the variables to be considered to boolean values (true when the residue exceeds the threshold, false otherwise), rather than to all possible real values.

Once the defects found, the problem is to isolate the defects, from signatures linking residues to faults, listed and validated by experts. Given the system model, the expected trace of a fault on the various analytical redundancy relations is designated as the signature of the fault ([PC91, Ger91, Fra96, CDD+00a]). The signatures of a set of faults are combined in the table of signatures. This matrix correlating analytical redundancy relations with faults, is the basis for the isolation of defects in the FDI approach. It is a model of bad behaviors.

For example, the boolean matrix in Figure 2.6 identifies four faults from three residues. It states, among other things, that the residue $r3$ is affected by the fault f_2 and is not affected by the other faults. In this matrix, the signature of the fault f_1 is given by the vector $(1, 0, 0)$.



	f1	f2	f3	f4
r1	1	0	0	1
r2	0	1	1	0
r3	0	1	0	0

signature of the fault f1

Figure 2.6: A matrix of fault signatures

The analytical redundancy relations are instantiated with the observed values. Then, the associated residues are calculated, providing an observed signature that can be compared to the theoretical signatures of the matrix. This comparison is a decision problem. Interpretation of the columns of the signature table is provided to generate a set of diagnoses, in terms of the faults represented in this table.

For example, with respect to the signature matrix of Figure 2.6, the signature of the observation defined by the vector $(1, 0, 0)$ states that the residue $r1$ is abnormal, indicating a fault, and the residues $r2$ and $r3$ correspond to "normal" values. It coincides with the (theoretical) signatures of the faults f_1 and f_4. This implies two possible diagnoses : $\{f_1\}$ and $\{f_4\}$.

Definition 2.6 *All diagnoses are given by faults whose signature is consistent with the observed signature.*

The diagnosis approach by analytical redundancy of the FDI community considers fault diagnosis as two separate tasks : fault detection and fault isolation. It is based on generating and evaluating a set of analytical redundancy relations derived from the system model and the available measurements (observations) coming from sensors. Fault detection involves computing the results (residues) of the relations by using the observed values, and checking if these residues exceed or not the associated thresholds. Finally, the residue evaluation defines the observed signature of the fault, so fault isolation consists in looking for the theoretical signature in the fault signature matrix that matches with the observed signature. And fault isolation requires considering all possible combinations of defects to capture multiple faults.

These diagnosis methods are used whenever the system can be represented by a mathematical model in the form of differential equations. There are other FDI methods, applicable when the system can be described as a DES, in other words when the system can be modeled by a finite state machine. For such systems, an observation is seen as a sequence of events and the diagnostic problem consists in comparing the observed event sequence with all the possible event sequences that can be generated by the model. The MBD approach for DES is explored in the next section.

2.4.3 Diagnosis of discrete event system

The diagnosis of DES is appeared in the 1990s with the work of Sampath [SSL+95, SSL+96]. The methods that emerge from this consist to infer the occurrences of fault events considered to be not observable, using the observable events generated by the system.

A DES is a dynamic system whose state can be modeled by a finite set of variables and whose evolution is discrete : time and system states continuously evolve, but the state changes are only considered as special moments, upon the occurrence of certain events. The state space of the DES model is discrete. The behavior of a DES can be modeled by means of transition systems (finite state machine, timed automata, Petri nets, timed Petri nets) or by means of process algebras as PEPA for example [CPR02, CPR00]. We consider in this section the diagnosis approaches for DES based on finite state machines [CL99].

In the context of DES diagnosis, two types of events are distinguished : the observable events, those whose occurrence can be observed, and the unobservable events that contain, among other, the faults. The DES model describes the normal and abnormal behavior of the system. The defects to be detected and diagnosed are known a priori and are represented by unobservable events. The goal of diagnosis is to detect and if possible to identify faults from the occurrence of observable events. The program that performs the diagnosis, called diagnoser, operates as a deterministic finite state machine that can be automatically built from the DES.

2.4.3.1 Diagnosis problem

Definition 2.7 *A model of DES is a finite state machine (X, Σ, T, x_0) where X is a finite set of states, Σ is a finite set of events, $T \subseteq (X x \Sigma x X)$ is a finite set of transitions and x_0 is the initial state of the system.*

When there is no constraint on the final state of the system, we consider that all states are final. Only paths of the finite state machine that depart from the initial state and terminate in a final state represent a possible behavior of the system. The event sequence on a path is called a trace. The observable trace of a trace is obtained by removing unobservable events from the trace. Thus, the finite state machine that models the DES generates all observable traces of the system, nomal ones (whitout fault event) as defective ones (containing fault events).

The diagnosis of DES is to study possible changes of a partially observable system to determine any malfunction by identifying the unobservable events that explain the observations.

Definition 2.8 *A diagnostic problem is defined by the system model (the finite state machine), the set of the observable events and observations in the form of a sequence of observed events.*

Definition 2.9 *A diagnosis is the set of possible faults whose occurrence is consistent with the observations, that is to say the set of faults belonging to a trace of the finite state machine whose the observable trace corresponds to the observation.*

The immediate diagnosis algorithm is based on searching all possible paths on the model, consistent with the provided observations, in order to find relatively simple information such as the occurrence of fault events or the current state of the system. The complexity of this search can be exponential as the number of paths in a finite state machine is often an exponential function of the size of the finite state machine.

2.4.3.2 Diagnosis algorithm

The diagnosis algorithm can be based on the construction of another finite state machine, called diagnoser, defined on the alphabet of the observable events, allowing to estimate the state of the system and to detect and to identfy the produced defects. The diagnoser, recognizing the sequence of observed events, yields an estimate of the current state of the system and of all occurring faults. Each state of the diagnoser is labeled by a set of pairs of the form (a state of the (model of) system, a partition of the set of faults) where the partition provides all faults found in the path reaching the state. For example, "$3\{f_1\}, 5\{\}, 5\{f_1\}, 8\{f_1, f_2\}$" is a suitable label for a diagnoser state. Three classes of states can be distinguished : the healthy states in which it is garanteed that no fault has occurred and for which all partitions are empty, the faulty states in which it is garanteed that a fault as occurred and for which no partition is empty, the ambiguous states in which some partitions are empty and others are not. The state "$3\{f_1\}, 5\{\}, 5\{f_1\}, 8\{f_1, f_2\}$" of a diagnoser is an example of ambiguous state.

To explain how a dignoser operates, let us consider the example of Figure 2.7. This example provides, in the left part of the figure, the finite state machine modeling a system, where the events f_1 and f_2 represent faults, o_1 and o_2 are the only observable events and $e1$ is an unobservable event. Starting from the initial state of the system model, since $e1$ and f_1 are not observable, it is impossible to know whether the system is in state 2 in the case of a normal behavior, or in the state 4 indicating the occurrence of a fault f_1. This situation is shown on the diagnoser, in the right part of the figure, by the initial state labeled "$2\{\}, 4\{f_1\}$". From the occurrence of the event o_1, the system model indicates that the system can be in state 1 indicating a normal behavior, or in state 4 indicating the occurrence of a fault f_1. This situation is shown on the diagnoser by the state, labeled "$1\{\}, 4\{f_1\}$", reached from the initial state "$2\{\}, 4\{f_1\}$" on the occurrence of the observable event o_1. This state "$1\{\}, 4\{f_1\}$" of the diagnoser cannot help to isolate the fault f_1 or to ensure the good behavior of the system. Only the occurrence of the event o_2 results in isolating the defect f_2 (reaching the faulty state "$3\{f_2\}$").

The diagnoser, built from the finite state machine of the system, can recognize sequences of observations, necessarily formed of the only observable events o_1 and o_2. Thus, the sequence of observations (o_1, o_2, o_2) is recognized by the diagnoser and explained by the occurrence of the fault f_2. For the observation sequence (o_1, o_1), it is possible either that the behavior of the system is not faulty, or that f_1 has occurred.

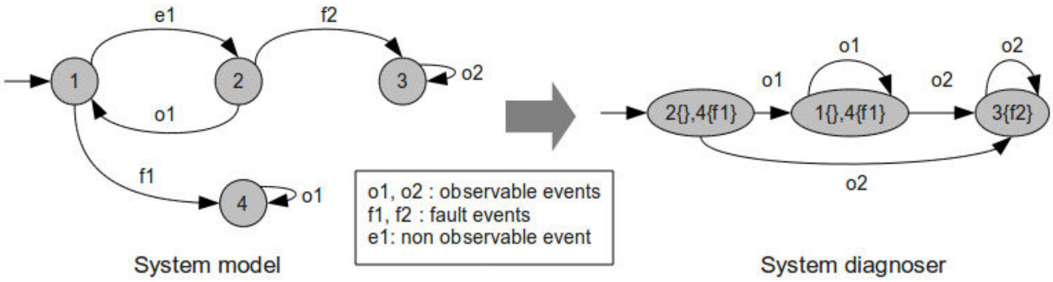


Figure 2.7: A diagnoser obtained from a finite state machine

The diagnoser is deterministic by construction. The algorithm for building the diagnoser from the system model is based on the list of the states reachable from a state by occurrence of an observable event and on fault propagation.

Algorithm 2.2 *Step 1*

Create the initial state X corresponding to the states that are directly reachable from the initial state of the system model, whithout occurrence of observable events.

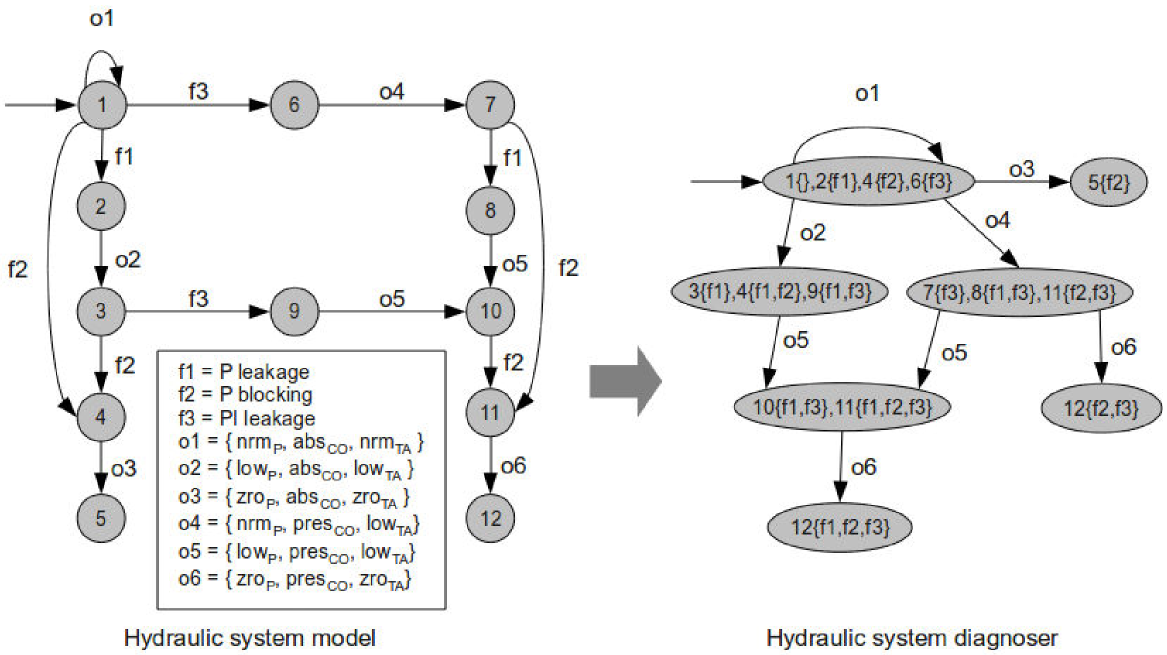
Step 2

For each observable event o, build a state Y of the diagnoser containing all the states reachable by o from X, by propagating failures.

Step 3

Consider instead of X, each newly created state Y, and return to step 2 until convergence.

Example 2.8 *Let us take again the example of the hydraulic system. The model of the system is given in the form of a finite state machine in the left part of Figure 2.8. This model incorporates normal and abnormal behaviors of the pump and the pipe. Il also expresses fault events using the* *unobservable events f_1, f_2 and f_3 representing a leak of the pump, a blockage of the pump and a leak of the pipe, respectively. Any indication of the three sensors is an observation. The normal behavior of the system is described by the initial state in which the system remains as long as the measures of the sensors correspond to the observation o_1. From this state, it is possible to model the occurrence of fauls lby the unobservable events f_1, f_2 and f_3. The diagnoser built from this model for diagnosing the hydraulic system is provided in the right part of Figure 2.8. Using this diagnoser, the observations (o_1, o_1, o_1, o_3) can be explained by the occurrence of the fault f_2, the observations $(o_1, o_1, o_1, o_4, o_6)$ can be explained by the ocurrence of the faults f_2 and f_3, the observation (o_4) is explained by the occurrence of the fault f_3 at least.*


o1
1
f3
6
o4
7
f1
2
8
f2
o2
o5
f2
3
f3
9
o5
10
f2
f2
4
11
o3
o6
5
12
f1 = P leakage
f2 = P blocking
f3 = Pl leakage
o1 = { nrm_P, abs_{CO}, nrm_{TA} }
o2 = { low_P, abs_{CO}, low_{TA} }
o3 = { zro_P, abs_{CO}, zro_{TA} }
o4 = { nrm_P, $pres_{CO}$, low_{TA}}
o5 = { low_P, $pres_{CO}$, low_{TA}}
o6 = { zro_P, $pres_{CO}$, zro_{TA}}
o1
1{},2{f1},4{f2},6{f3}
o3
5{f2}
o2
o4
3{f1},4{f1,f2},9{f1,f3}
7{f3},8{f1,f3},11{f2,f3}
o5
o5
o6
10{f1,f3},11{f1,f2,f3}
12{f2,f3}
o6
12{f1,f2,f3}

Hydraulic system model

Hydraulic system diagnoser

Figure 2.8: A diagnoser for the hydraulic system

The principle of MBD of DES can be summarized as follows. In a first step, a model of normal and abnormal behavior of the DES is constructed in the form of a finite state machine, including faults as events. Intuitively, a fault, associated with the occurrence of an unobservable event, is followed by a finite sequence of observable events that does not occur in the absence of the fault. From this model, the diagnosis tool, called diagnoser, is built, implementing a deterministic finite state machine. The diagnoser is used during system functioning. From a sequence of observable events generated by the system, it provides an estimate of the system state and of the faults affecting its behavior. This approach assumes that the observations are totally ordered. The difficulty of the task lies mainly in the fact that the system is partially observable: in the model of Figure 7, the sequences (o_1, f_1, o_1) and (o_1, e_1, o_1) correspond to the same observable trace (o_1, o_1).

The diagnoser approach is based on prior computation of the set of possible behavior paths. This approach is limited by the size of the diagnoser. When the number of states of the diagnoser is large, it is expensive in memory requirements or execution time. Although decentralized or distributed solutions have been developed to overcome this difficulty [PCR02, GL03, LDM09], another approach [Tri02, GCL05] consists in calculating only the paths compatible with observations while the system is functioning. It is to determine if there are paths on the system model, normal or faulty, that are compatible with the observations.

2.4.4 Temporal MBD

A dynamic system is one that evolves over time, and so that its future only depends on phenomena of the past and the present. In other words, its state at a time depends on its past states, it has a memory of its past states [CDPTM13]. In addition, its behavior is deterministic : one single future state is possible at any time. This makes possible to predict (calculate) its behavior over time. It is usual to distinguish between DES whose states change on the occurrence of events, and continuous dynamic systems whose states continuously change over time, usually modeled with differential equations or represented by discretizing the time and the system states.

The MBD methods presented in the previous sections say nothing explicit about time representation and how the diagnosis process can take into account the temporal dimension of dynamic systems. In particular, [Str91] and [DK03] have shown that the logical theory of diagnosis proposed by Reiter [Rei87] is hard to apply for dynamic systems. Thus, some of the temporal MBD approaches are temporal extensions of consistency-based atemporal diagnosis or of the abductive atemporal diagnosis [BCTTD98]. When observations are not dated or are not associated with any temporal information, only the precedence of the observations may be taken into account. In the system model, time is considered qualitative through the occurrence order of the observations. And the diagnosis algorithm considers the time in the same way, that is to say, from a qualitative point of view. For example, a DES diagnoser (section 2.4.3.2) built from a model that only considers the occurrence order of events, attempts to recognize a sequence of observable events that may correspond to a fault. However, with the model of Figure 2.9 that contains the observable events o_1 and o_2, the unobservable faulty event f_1, and the unobservable normal event e_1, the behaviors (o_1, f_1, o_2) and (o_1, e_1, o_2) correspond to the same observable trace (o_1, o_2) and the faulty behavior (the one including f_1) will not be able to be distinguished from the normal behavior.

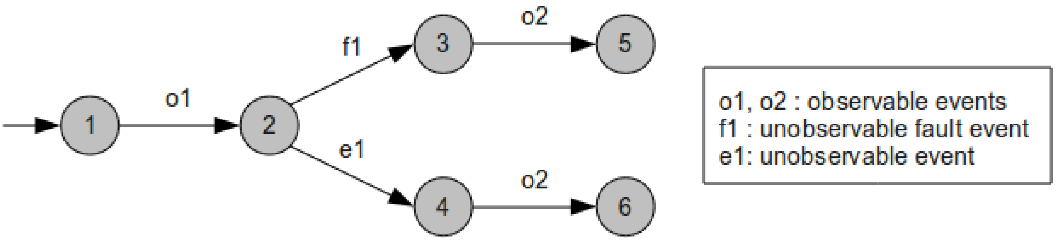


Figure 2.9: A finite state machine modeling a DES

Although some studies, mentioned in [SSD97], have shown that, in some cases, it is possible to make a diagnosis of dynamic systems ignoring the temporal problems, exploiting the temporal aspect of the dynamic system can increase the discriminatory capacities of the diagnosis process. Occurrence dates of the observations and temporal constraints between observations may be essential to discriminate defects and thus to achieve the diagnosis task. In this case, time must be explicitly taken into account in a quantitative manner. It involves adding a time dimension to the system model and using appropriate algorithms, depending on the chosen representation of time. Taking the temporal dimension into account necessarily makes diagnosis significantly more complex. At the same time, there is no common general principles that can be found within the MBD methods that incorporate an explicit time management [BCTTD98, CDPTM13].

In the particular case of DES, the temporal aspect may be introduced into the model to represent explicitly and quantitatively the temporal relationships linking occurrences of events. For this, the diagnosis methods for DES rely on temporized models based on transition systems : timed automata [AD94], time or timed Petri nets [Mer74, Ram74], chronicles [Dou94], etc. And observations are commonly assumed to be a collection of timed observable events.

Timed automata, introduced by Alur and Dill in the 1990s [AD94], extend classical automata with a set of clocks, that are represented by real-valued variables and that increase synchronously with time, and with constraints associated with every transition, specifying when (i.e. for which values of the clocks) the transition can be performed. With these clocks, it becomes possible to express constraints between two transitions. For example, the timed automaton in Figure 2.10 models two kinds of behaviors of a DES. In faulty behaviors, o_1 is followed by o_2 within a delay greater to 3 time units, while in normal behaviors, the delay between the occurrences of o_1 and o_2 must be less than or equal to 3 time units. Thus, a diagnoser observing the sequence of events (o_1, o_2) and measuring their inter-arrival delay can decide whether a fault occurred or not. [Tri02] proposes an extension of the diagnosis process for DES using a timed automaton to model the system distinguishing between normal and abnormal states, such as the diagnosis algorithm is able from this timed automaton, to estimate the system state following an occurrence of an observable event and to trigger an alarm when a fault is detected. [DYHAA06] proposes a diagnosis approach based on a diagnoser previously built from a timed automaton.

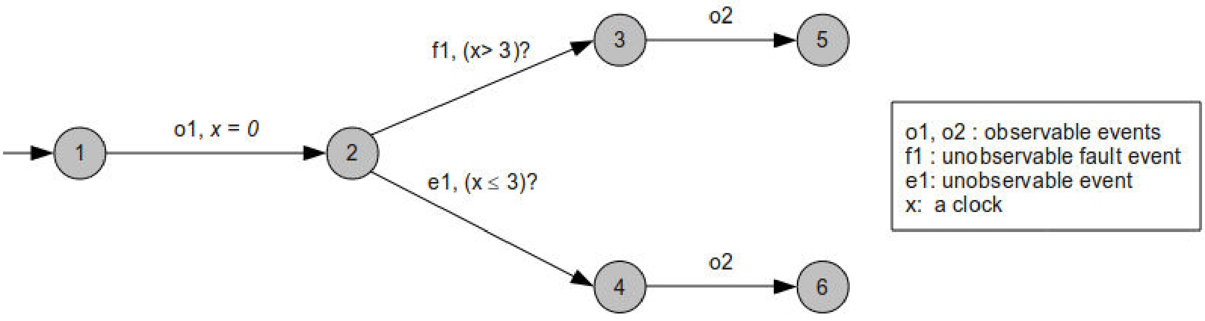


Figure 2.10: A timed automaton with one clock modeling a DES

In classical Petri nets, time is represented by the ordered sequence of events as transitions firing. Classical Petri nets have been extended to take time into account explicitly. Time is considered either as a delay (time Petri nets) or as intervals of dates (timed Petri nets). In both cases, time annotation can be attached to places or to transitions. In the same way as diagnosis based on timed automata, the diagnosis algorithm not only infers causalities between observed events, but also calculates the possible dates of occurrence of each observed event. The diagnosis process is based on the unfolding technique of Petri net.

A chronicle defines a partially ordered set of events constrained by temporal relationships. A similar formalism is those of causal temporal signature defined in [TCG90]. Chronicles [Dou94] are patterns of sequences of timed events, representing a possible abnormal situation, to be recognized by the diagnosis algorithm as the system is functioning. Here, the issue is one of incremental diagnosis by temporal decomposition [GCL05]. The objective is to reason effectively with incomplete knowledge about the system evolution. The method relies on calculating temporal windows of the observations that are expected by the current chronicles. For example, when the constraint of the chronicle indicates that the expected occurrences of two different observed events must be spaced by 5 to 10 time units, the observation of the occurrence of the first event at time t determines the occurrence of the second event between t+5 and t+10. When all the events of the chronicle can be mapped to the timed observations, the chronicle is recognized. The recognition of a chronicle is associated with the manifestation of a fault. Once the set of chronicles is described by the expert, the diagnosis process receives a stream of dated observations it seeks to match with the chronicles. Chronicles may be modeled by a time Petri net or by a graph of temporal constraints. The chronicle formalism has been used for fault diagnosis in several applications [CD00, CLGR$^+$07, GD04]. In [HP00], the authors uses patterns of sequences of events, called templates, that characterize normal behaviors of the system instead of faulty ones. A template which defines a triggering event e_1 and a consequent event e_2 with a temporal window $[t_1, t_2]$ indicates that if e_1 is observed at time t then e_2 is assumed to be observed between $t + t_1$ and $t + t_2$, otherwise a fault is detected. Templates are set of constraints on the occurrence times of events. The template approach may be seen as a diagnosis approach that uses a timed automaton without clocks, but where time intervals are associated with states [Tri02]. Finally, each pattern, whether a causal temporal signature, a chronicle or a template, is derived from expert knowledge and may be built separately regardless of other patterns. Also, it is necessary to check the consistency of the set of patterns to ensure the completeness and the determinism of the recognition process.

2.5 Multi-modeling diagnosis

The diversity of the approaches that have been proposed for system diagnosis is the result of various contexts associated to the nature and the characteristics of the target applications. The specific implemented diagnosis strategies depend both on the type of information available about the system, and on the type of defects to be detected. Of all the methods, MBD methods play an important part. For these methods, the modeling step is the crucial step of the diagnosis problem. Since the model is directly used as a reference for fault detection, diagnosis quality depends on the quality of this model and the model should be selected based on its ability to reveal faults.

The models that are used for diagnosis exploit three types of knowledge (cf. figure 2.11). [CGTT93] distinguish them in functional, behavioral and structural models. The structural model of the system describes the components that are needed to design the system, with their interconnections. The behavioral model describes the behavior of the components, how the components interact with each other through relationships involving physical quantities. The third dimension is functional. It defines the role of the components for the diagnosis purpose. For example, the function of the tank component of an hydraulic system is to store water, the function of the pump is to convert electrical power into hydraulic power. The defects exhibit, in the expert opinion, significant effects in terms of observations, while, from the point of vue of the user of the system, they reflect malfunctions on physical units that must be repaired or replaced. The link between the expert's perspective and the user's one is defined by the functional dimension. The function is a subjective interpretation of the expert knowledge, to make it understandable by the user, that defines the relationship between the behavior and the objectives assigned to the system by its designer. Structural and behavioral knowledge is of fundamental knowledge, while functional knowledge is interpretative knowledge.

The aim of a multi-modeling approach is clearly to exploit all the fundamental and interpretative knowledge available about the system, to achieve a complex reasoning task, namely the diagnosis task. For some authors, the different types of knowledge can be mixed within the same model. On the other hand, for [CGTT93], the structural, behavioral or functional models are separated and need to be interconnected : links exist that can switch from one knowledge type to another and these links must be precisely described.

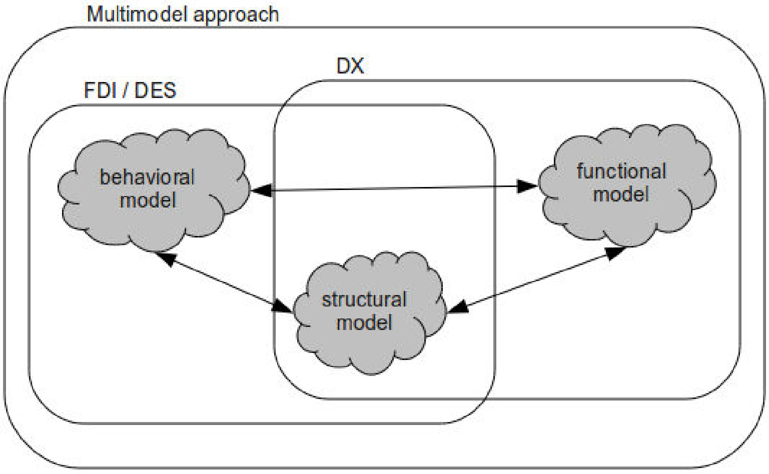


Figure 2.11: The multi-modeling approach

The aim problem for the diagnosis of dynamic systems comes from the acquisition and the representation of the underlying temporal knowledge. These tasks are difficult since temporal knowledge is often mixed with other types of knowledge. This justifies multi model based diagnosis that put emphasis on the separation of the different types of knowledge in different models, in order to simplify both the knowledge representation and the reasoning on the different models [ML07, LGM07, LGMC08].

2.6 Conclusion

Generally speaking, to diagnose a system consists in a reasoning aiming at explaining the available *observations* with a *model* of the system according to an adequate *problem solving method.* Formal theories of diagnosis have been elaborated since the last 50 years in order to build programs that facilitate this reasoning. The possibility for a computer to achieve such a diagnosis reasoning depends on the manner*observations*, *models* and *problem solving methods* are formalized.

This chapter presents the basis of the three main categories of diagnosis approaches (DX, FDI and DES for short), with the example of an hydraulic system described in [CPR00]. The aim is to elicit the common concepts and the specificities of each of these main approaches, and to emphasis on the particular difficulties when diagnosing complex dynamic system where the temporal aspects play a crucial role. In particular, this chapter presents the Multi-Modeling Based Diagnosis (MMBD) as an attempt to deal with complex dynamic processes because such processes are difficult to model. As Figure 2.11 aims to illustrate, the goal of the MMBD approaches is to try to take into account the multiple aspects of the knowledge that is required to diagnose such processes and then to take advantages of the huge scientific background developed by the DX, the FDI and the DES research communities.

The next chapters of this document proposes an a new MMBD approach called TOM4D for Timed Observation Modeling for Diagnosis and a solver algorithm called TOM4E (Timed Observation Manager for Explanation) designed to diagnose networks of dynamic processes. The TOM4D MMBD approach is *new* because it is the only diagnosis approach that is based on the Timed Observation Theory (TOT) proposed by Le Goc in 2006 [LG06], where the timestamps of the data play a central role. This mathematical theory combines and extends the Markov Chain and the Poisson Process Theories [CL99], the Theory of Communication of Shannon [Sha84], the Logical Theory of Diagnosis [Rei87] and the Method of Abstraction of Floridi [Flo08]. Up to our knowledge, the TOT is the only mathematical theory that has been designed to propose a unique framework to model networks of continuous time dynamic processes at different levels of abstraction and granularity.

The mains applications of the TOT is the TOM4D Knowledge Engineering methodology presented in this document, a Knowledge Discovery in Database process called TOM4L (Timed Observation Mining for Learning, [LGBG05, BLG10, LGA12]) and an operational method of abstraction called TOM4A (Timed Observation Method for Abstraction, [Pom12a, PMAP12]). The main advantages of these applications is that, being based on the same mathematical theory, TOM4D, TOML and TOM4A are compatible together so that their results can easily be merged in order to build and to validate models from the available real-world timed data (cf. [PLG14] for an illustration).

The next chapter of this document introduces the basic concepts of the TOT. The TOM4D methodology version that is proposed in the following chapter generalizes the version provided in PHD's Pomponion document [Pom12a] and extends it to the modeling of networks of dynamic processes. The next chapter is then dedicated to the description of an adequate diagnosis problem solver for TOM4D, the TOM4E algorithm. Finally, the ability of the TOM4D MMBD approach is shown with its application on a real world dynamic process, the Sapins french hydraulic dam. This explains why the hydraulic example of [CPR00] has been used in this chapter and will be used all along the next chapters in order to facilitate the reading of this document.

CHAPTER 3

Introduction to the Timed Observation Theory

The Timed Observations Theory (TOT, [LG06]) provides a general mathematical framework for modeling dynamic processes from timed data. This theory combines and extends the Markov Chain Theory, the Poisson Process Theory, the Theory of Communication of Shannon [Sha84], the Logical Theory of Diagnosis [Dag01] and the Method of Abstraction of Floridi [Flo08].

This chapter aims at introducing the concepts of the TOT that are required in order to model a dynamic process and the diagnosis reasoning using it. To this aim, the chapter is made with to sections: the first one introduces the mathematical objects of the TOT that will constitutes the foundation of the conceptual modeling framework of our contributions. This later is described in the second section.

The complete mathematical theory of timed observation can be found in [LG06]. Applications in divreses scientific domain can be found in the PhD of P. Bouche [Bou05] for the Discrete Event System domain, the PHD of N. Benayadi [Ben10] for the Data Mining domain, the PHD of A. Adhab [Ahb10] for the Bayesian Networks domain, and the PHD L. Pomponio [Pom12b] for the Knowledge Engineering Domain. All over this section, the Reader will be invited to refer to the main papers that provide details about the developments of the TOT.

3.1 Mathematical Framework of the TOT

The Timed Observations Theory defines a *dynamic process* as an arbitrarily constituted set $X(t)$ = $\{x_1(t), ..., x_n(t)\}$ of timed functions $x_i(t)$ of continuous time t. The set $X(t)$ of functions implicitly defines a set $X = \{x_1, x_2, ..., x_n\}$ of n variable names x_i.

According to the Timed Observation Theory (TOT), a dynamic process $X(t)$ is said to be *observed* by a program Θ when this latter aims at writing *timed messages* describing the modifications over time of the functions $x_i(t)$ of $X(t)$. Such timed messages can be alarms, warnings, reporting events or simple communication messages (sms for examples) that are sent to the environment of the program Θ. A timed message is a sequence of characters that can be recorded in a memory (i.e. a database or a data log). For example, the timed message “`1999/04/09 08:33:00, The pipe outflow is low`” can be an alarm. Such a timed message can easily be recorded in a database (often called a data log)

The TOT considers that the structure of such a message is a simple pair (*timeStamp*, *text*) where *text* is a *constant* denoted δ_i and *timeStamp* is the value of an *index* denoted t_k. The time stamp t_k represents the time of the message that can be the emission time, the reception time or any time stamp associated with the text of the message. Consequently, an extraction of m timed messages from such a memory constitutes a set $\Delta = \{\delta_i\}$ of m_Δ constants δ_i and a set $\Gamma = \{t_k\}$ of m time stamps t_k. Generally speaking, there is less constants than time stamps (i.e. $m_\Delta \leq m$).

Θ is said to be :

- *Parametrized* with two sets, the set of constants $\Delta = \{\delta_i\}$ and the set of variable names $X = \{x_i\}$. A parametrized program is denoted $\Theta(X, \Delta)$.
- *Applied* on a set $X(t) = \{x_i(t)\}$ of timed functions. The choice of $X(t)$ is out of the scope of the program.
- An *abstract observer* if the way the program Θ has been implemented is not known. It can be either a software component or a human. This situation has to be considered as the usual situation.

In the suite of the document, the term *program* will be used to denote either the program Θ or the parametrized program $\Theta(X, \Delta)$.

3.1.1 Timed Observation

The aim of the TOT is to model *observed processes*:

Definition 3.1 *Observed Process*
Let $X(t) = \{x_i(t)\}$, i=1...n, be a finite set of time functions $x_i(t)$; let $X = \{x_i\}$, i=1...n, be the corresponding finite set of variable names x_i; let $\Delta = \{\delta_j\}$, j=1...m, be a finite set of constant values δ_j; let $\Theta(X, \Delta)$ be a program observing the evolution of the functions of $X(t)$.

The couple $(X(t), \Theta(X, \Delta))$ is an observed process.

To this aim, the TOT defines a *timed observation* to provide a *meaning* to a timed message:

Definition 3.2 *Timed Observation*
Let $X(t) = \{x_i(t)\}_{i=1\ldots n}$ be a set of time functions describing the evolution of a process that is observed by a program Θ; let $\Gamma = \{t_k\}_{t_k \in \Re}$ be a set of arbitrary time instants in which Θ observes the functions; let $\theta(x_\theta, \delta_\theta, t_\theta)$ be a predicate implicitly determined by Θ; and, let Δ be a set of constant values.

A timed observation $(\delta, t_k) \in \Delta \times \Gamma$ made on the time function $x_i(t)$ is the assignation of values x_i, δ and t_k to the predicate $\theta(x_\theta, \delta_\theta, t_\theta)$ such that $\theta(x_i, \delta, t_k)$.

Technically, (δ, t_k) (or $o(t_k)$) denotes a *record* of a database. The assigned predicate $\theta(x_i, \delta, t_k)$ represents the *meaning*, the *interpretation* of the record. But, by misuse of language, (δ, t_k) (or $o(t_k)$) are usually called a *timed observation.*

As an illustration of this definition, let us consider a database containing records that are similar to the message "`1999/04/09 08:33:00, The pipe outflow is low`". The records are of the form "`yyyy/mm/dd hh mm ss, text`" where:

- `yyyy/mm/dd hh mm ss` is a time stamp
- "," is a separator character,
- and `text` is a sequence of characters.

Thus, the message "`1999/04/09 08:33:00, The pipe outflow is low`" will be represented with a timed observation (δ_i, t_k) where $\delta_i =$ `The pipe outflow is low` and $t_k =$ `1999/04/09 08:33:00`:

- $(\delta_i, t_k) = ($`The pipe outflow is low, 1999/04/09 08:33:00`$)$.

One of the most fondamental point of the TOT is to understand that, *without any knowledge about the assigned predicate* $\theta(x_i, \delta, t_k)$, a timed observation (δ, t_k) has *no meaning.* For example, according to the TOT the text "*The pipe outflow is low*" has no meaning: it is only a sequence of characters. The next section illustrates this point.

3.1.2 Spatial Discretization Principle

Let us consider a particular program $\theta_i(\{x_i\}, \{\delta_i\})$ of an observed process $(x_i(t), \theta_i(\{x_i\}, \{\delta_i\}))$ made of only one time function $x_i(t)$. Let us suppose that the specification of such a program is based on the generic rule 3.1 which refers to a threshold value $\Psi_j \in \Re$ and two immediately successive values $x_i(t_{k-1}) \in \Re$ and $x_i(t_k) \in \Re$.

$$x_i(t_{k-1}) < \Psi_j \wedge x_i(t_k) \geq \Psi_j \Rightarrow write((\delta_i, t_k)) \tag{3.1}$$

In this rule:

- $x_i(t_{k-1})$ and $x_i(t_k)$ are two immediately successive values of the continuous time function $x_i(t)$,

- (δ_i, t_k) is a timed message,

- $x_i(t_{k-1}) < \Psi_j \wedge x_i(t_k) \geq \Psi_j$ specifies a particular predicate denoted θ_i,

- $write((\delta_i, t_k))$ denote the action of recording a timed message in a memory.

In other words, a timed observation (δ, t_k) is the *execution trace* of the program θ_i.

The use of such a rule is illustrated in figure 3.1 where two thresholds Ψ_1 and Ψ_2 and the four rules are used:

$$\text{Rule 1: } x_i(t_{k-1}) < \Psi_2 \wedge x_i(t_k) \geq \Psi_2 \Rightarrow write((high, t_k)) \tag{3.2}$$

$$\text{Rule 2: } x_i(t_{k-1}) \geq \Psi_2 \wedge x_i(t_k) < \Psi_2 \Rightarrow write((normal, t_k)) \tag{3.3}$$

$$\text{Rule 3: } x_i(t_{k-1}) < \Psi_1 \wedge x_i(t_k) \geq \Psi_1 \Rightarrow write((normal, t_k)) \tag{3.4}$$

$$\text{Rule 4: } x_i(t_{k-1}) \geq \Psi_1 \wedge x_i(t_k) < \Psi_1 \Rightarrow write((low, t_k)) \tag{3.5}$$

Let us suppose that these four rules are implemented in 4 programs, respectively $\theta_1(\{x_i\}, \{high\})$, $\theta_2(\{x_i\}, \{normal\})$, $\theta_3(\{x_i\}, \{normal\})$ and $\theta_4(\{x_i\}, \{low\})$. Let us suppose also that these programs have been combined in a unique program $\Theta(\{x_i\}, \{high, normal, low\})$:

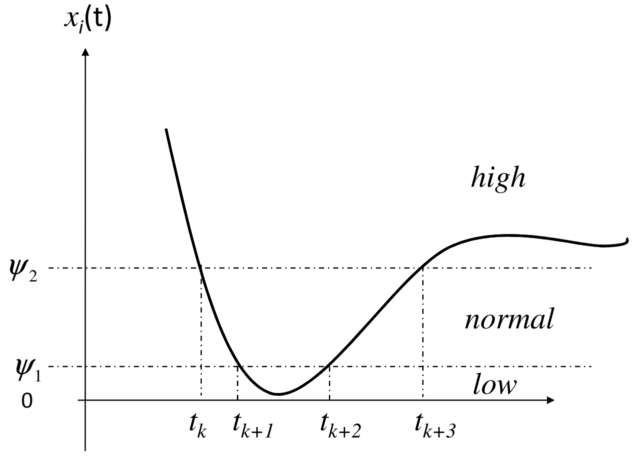


Figure 3.1: Spatial Discretization of the time function $x_i(t)$ with two thresholds

$$\begin{gathered}\Theta(\{x_i\},\{high, normal, low\}) = \\ \theta_1(\{x_i(t)\},\{high\}) \cup \theta_2(\{x_i(t)\},\{normal\}) \cup \theta_3(\{x_i(t)\},\{normal\}) \cup \theta_4(\{x_i(t)\},\{low\})\end{gathered} \tag{3.6}$$

The observed process $(x_i(t), \Theta(\{x_i\},\{high, normal, low\})$ allows the program Θ to produce the following sequence ω of timed observation with the time function $x_i(t)$ of figure 3.1:

- $\omega = \{(normal, t_k),\ (low, t_{k+1}),\ (normal, t_{k+2}),\ (high, t_{k+3})\}$.

The sequence ω is the execution trace of the program $\Theta(\{x_i\},\{high, normal, low\})$ when it observes the time function $x_i(t)$. According to the definition 3.2, such a sequence defines a set $\Gamma = \{t_k, t_{k+1}, t_{k+2}, t_{k+3}\}$ containing four time stamps. Because the time function is defined over $\Re$, the duration between two immediately following time stamps is random (i.e. $(t_{k+1} - t_k) \neq (t_{k+3} - t_{k+2})$ for example). Γ is then a *stochastic clock.*

Because a predicate θ_i implemented in a program $\theta_i(\{x_i\},\{\delta_i\})$ can have a very complex meaning requiring a lot of computation, the general form of the spatial discretization principle is based on the following rule:

$$\theta(x_i, \delta_i, t_k) \Rightarrow write(o(t_k)) \tag{3.7}$$

An important point at this stage is that the definition 3.2 of a timed observation shows that the link between a time function $x_i(t)$, a variable name x_i, the value of a constant δ_i and a timed observation $o(t_k)$ is made by the program $\theta_i(\{x_i\},\{\delta_i\})$ that implements an instantiation of the general rule 3.7:

- The relation between a time function $x_i(t)$ and a variable name x_i is made through the application of the program $\theta_i(\{x_i\}, \{\delta_i\})$ on $x_i(t)$.
- The variable name x_i is defined for the predicate θ_i: the function $x_i(t)$, the variable name x_i and the constant δ_i are *independent*.
- The constant δ_i of a timed observation $o(t_k) \equiv (\delta_i, t_k)$ is not directly linked with the predicate θ: δ_i is simply *linked* to θ by the *code*.

A constant δ_i is then clearly a sequence of characters that has no meaning in it-self: a pair (δ_i, t_k) is only a record in a database. To provide a meaning to this pair, the reader *must have* an *interpretation model*.

3.1.3 Semantic of a Timed Observation

As an illustration of this important point, let us consider the equations 3.3 and 3.4. The same constant "*normal*" is used for two different predicates (i.e. two programs $\theta_2(\{x_i\}, \{normal\})$ and $\theta_3(\{x_i\}, \{normal\})$):

- θ_2: $x_i(t_{k-1}) \geq \Psi_2 \wedge x_i(t_k) < \Psi_2$
- θ_3: $x_i(t_{k-1}) < \Psi_1 \wedge x_i(t_k) \geq \Psi_1$

The constant "*normal*" has then *two* meanings:

- θ_2: *normal* means that the values of the time function $x_i(t)$ left the interval $[\Psi_2, +\infty[$ to enter the interval $]-\infty, \Psi_2[$.
- θ_3: *normal* means that the values of the time function $x_i(t)$ left the interval $]-\infty, \Psi_1[$ to enter the interval $[\Psi_1, \Psi_2]$.

If the two programs $\theta_2(\{x_i\}, \{normal\})$ and $\theta_3(\{x_i\}, \{normal\})$ have no error, a unique and common meaning to the sequence of characters "*normal*" can be provided: the values time function $x_i(t)$ are in the range *normal* corresponding to the interval $[\Psi_1, \Psi_2[$. As a consequence, the 4 rules 3.2 to 3.5 defines 3 ranges for the values of the time function $x_i(t)$:

- *high* range: $[\Psi_2, +\infty[$,
- *normal* range: $[\Psi_1, \Psi_2[$,
- *low* range: $]-\infty, \Psi_1[$.

This example shows a fundamental point that is highlighted by the TOT: **without some knowledge about the program θ_i that write it, a timed message contained in a database has no meaning in itself**.

So, the timed message "`1999/04/09 08:33:00, The pipe outflow is low`" has no meaning because the program that has writen this message has not been described. In other words, the sequence of characters *The pipe outflow is low* is a constant that can be rewritten δ_i without changing anything.

But if it is know that Figure 3.1 illustrates the water level in the pipe component of the didactic system presented in the precedent section, then a particular meaning can be given to the timed message "`1999/04/09 08:33:00, The pipe outflow is low`" and so to the corresponding timed observation $o(t_k) \equiv (\delta_i, t_k) =$ (`The pipe outflow is low, 1999/04/09 08:33:00`):

- The time function $x_i(t)$ represents the level of the outflow in the pipe.
- The sequence of characters *low* can be associated with the rule 3.5. and then with the predicate θ_4 (i.e. $x_i(t_{k-1}) \geq \Psi_1 \wedge x_i(t_k) < \Psi_1$).
- The variable name x_i denotes the values $x_i(t)$ takes at the time stamps t_k, t_{k+1}, t_{k+2} and t_{k+3}.

As a consequence, the sequence of characters *The pipe outflow is low* means that the predicate θ_4 has been satisfied at time $t_{k+1} =$"`1999/04/09 08:33:00`". But when no knowledge is available about the way a pair (δ, t_k) has been recorded in a database, the definition 3.2 allows to infer that:

Theorem 3.1 *Interpretation*
Given a timed observation $o(k) \equiv (\delta, t_k)$, the following propositions are true:

1. *$o(k)$ has been written by an abstract program $\theta(\{x\}, \{\delta\})$ that defines a ternary predicate $\theta(x_\theta, \delta_\theta, t_\theta)$.*
2. *There exist a time function $x(t)$ which has been observed by the abstract program $\theta(\{x\}, \{\delta\})$.*
3. *At time $t = t_k$, the time function $x(t)$ satisfied the constraints of the predicate $\theta(x_\theta, \delta_\theta, t_\theta)$.*
4. *The meaning of $o(k) \equiv (\delta, t_k)$ is the assignation $\theta(x, \delta, t_k)$.*

When considering the Spatial Discretization Principle (cf. equation 3.1), the assignation $\theta(x, \delta, t_k)$ can have one of the three following interpretations:

- $EQUAL(x, \delta, t_k)$: "At time t_k, x **is equal to** δ"
- $IS(x, \delta, t_k)$: "At time t_k, x **is** δ"
- $BELONGS(x, \delta, t_k)$: "At time t_k, the values of $x(t)$ **belongs to** a range denoted δ"

These three interpretations are clearly misuses of language because δ is a constant taken from an arbitrary made set Δ and the definition domain of the function $x(t)$ is the set $\Re$ of the real numbers. For example, according to the theorem 3.1, the timed observation (`The pipe outflow is low, 1999/04/09 08:33:00`) can be interpreted as :

1. *At time 08:33:00 the 1999/04/09, the outflow of the pipe is equal to low.*
2. *At time 08:33:00 the 1999/04/09, the outflow of the pipe is low.*
3. *At time 08:33:00 the 1999/04/09, the outflow of the pipe belongs to the low range.*

The first interpretation is the most usually used because it corresponds to the abuse of language $x(08{:}33{:}00\ 1999/04/09) = low$ which has the form of a classical formula:

$$x(t_k) = \delta \tag{3.8}$$

Generally speaking, in practice, a predicate $\theta(x_\theta, \delta_\theta, t_\theta)$ is satisfied when the time function $x(t)$ matches against a behavior model [LG04] that can be as simple as the switch of an interrupter or, requiring complex techniques, such as signal processing techniques for artificial vision. The precise meaning of such a predicate can be very complex and very difficult to detail. This explains why most every experts commonly use the abuse of language of the equation 3.8.

It is noteworthy that a program can have errors: (δ, t_k) could be written in a database from the assertion $\theta(x, \delta, t_k)$ although the time function $x(t)$ not "really matches" the semantic of this predicate.

3.1.4 Observation Class

When considering a timed observations $o(k) \equiv (\delta, t_k)$, the first interpretation (and the equation 3.8) explains the fact that an expert establishes immediately (and often unconsciously) a relation between the constant δ and a variable name x.

This cognitive phenomena being so important and so natural, the TOT defines it with the notion *observation class*:

Definition 3.3 *Observation Class*
Let $X(t) = \{x_i(t)\}_{i=1...n}$ be a set of time functions that are observed by an abstract program $\theta(X, \Delta)$ where $\Delta = \{\delta_j\}_{j=1...m}$ is the set of all the constants the abstract program can use and $X = \{x_i\}_{i=1...n}$ is the set of variable names corresponding to $X(t)$.

$\forall i \in [1, n]$, $\forall j \in [1, m]$ and $\forall k \in \mathbb{N}$, an observation class $C_k = \{..., (x_i, \delta_j), ...\}$ is a subset of $X \times \Delta$.

An observation class is then any set of pairs (x_i, δ_j) associating a variable name x_i with a constant δ_j. Such a definition establishes an *explicit* link between a constant and a variable name. Any association can be made, but the simplest way, and the most used, is to associate a variable x_i to each constant δ_j (i.e. establishing a mapping $\delta_j \mapsto x_i$ for each $\delta_i \in \Delta$) and to define all the observation classes with singletons $C_i = \{(x_i\delta_j)\}$, that is to say where the pair (x_i, δ_j) is the unique element the set C_i. In that case, the following theorem can be applied:

Theorem 3.2 *Class Occurrences*
Let $\Delta = \{\delta_j\}_{j=1...m}$ be a set of m constants δ_i; let $X = \{x_i\}_{i=1...n}$ be a set of n variables names x_i so that $n \leq m$; let $C = \{C_j\}_{j=1...m}$ be a set of m singletons $C_j = \{(x_i, \delta_j)\}$.

Any timed observation $o(tk) \equiv (\delta_j, t_k)$ written by a program $\theta(X, \Delta)$ is an occurrence of an observation class $C_j = \{(x_i, \delta_j)\}$.

This theorem and the theorem 3.1 leads to define a mapping from the set of constants Δ to the set of variable names X. This simplifies strongly the situation when the variable names are unknown: it is always possible to map an abstract variable ϕ_j to each constant δ_j that appears in a sequence ω of timed observation (i.e. in an extraction of a database). This is done with the construction of a set $C = \{C_i\}$ of observation classes $C_i = \{(\phi_i, \delta_i)\}$ where each C_i is a singleton. In that case:

$$o(tk) \equiv (\delta_i, t_k) \equiv C_i(t_k) \tag{3.9}$$

In other words, when defining observation classes as singletons, a program $\Theta(X, \Delta)$ observing a process $X(t)$ writes occurrences $C_j(t_k)$ of observation classes C_j and the equation 3.7 can then be written in its most abstract form:

$$\theta_i(x_i, \delta_j, t_k) \Rightarrow write(C_j(t_k)) \tag{3.10}$$

For example, the program $\Theta(\{x_i\}, \{high, normal, low\})$ associated with the time function $x_i(t)$ of figure 3.1 produces the sequence ω of timed observation $\omega = ((normal, t_k)$, (low, t_{k+1}), $(normal, t_{k+2})$, $(high, t_{k+3}))$. This sequence allows to define the set C of observation classes containing the followings classes:

- $C_1 = \{(x_i, low)\}$
- $C_2 = \{(x_i, normal)\}$
- $C_3 = \{(x_i, high)\}$

In that case, the preceding sequence can then be written:

- $\omega = \{C_2(t_k), C_1(t_{k+1}), C_2(t_{k+2}), C_3(t_{k+3})\}$.

This example shows the following important points of the TOT:

- The set C contains *three* classes when the program $\Theta(\{x_i\}, \{high, normal, low\})$ is made with *four* programs (cf. equation 3.6). This shows that a program $\Theta(X, \Delta)$ can be made with an arbitrary number of concrete programs that can be as simple or as complex as necessary.
- In the sequence ω, the observation class C_2 has two occurrences: $C_2(t_k)$ and $C_2(t_{k+2})$. When considering the figure 3.1 and the equation 3.6, it is obvious that the two occurrences haven't been written by the same program: $\theta_2(\{x_i\}, \{normal\})$ for $C_2(t_k)$ and $\theta_3(\{x_i\}, \{normal\})$ for $C_2(t_{k+2})$. But when interpreting these timed observation according to the equation 3.8, the two programs can be confused in an abstract one $\theta_{23}(\{x_i\}, \{normal\})$ $= \theta_2(\{x_i\}, \{normal\}) \cup \theta_3(\{x_i\}, \{normal\})$.
- The definition 3.3 allows to partition the sequence ω in three sequences, each of them being associated with an observation class:
  - $\omega_1 = \{C_1(t_{k+1})\}$, $\omega_2 = \{C_2(t_k), C_2(t_{k+2})\}$, $\omega_3 = \{C_3(t_{k+3})\}$
  - $\omega = \omega_1 \cup \omega_2 \cup \omega_3$
  - $\omega_1 \cap \omega_2 \cap \omega_3 = \Phi$
- As a consequence, the set $\Gamma = \{t_k, t_{k+1}, t_{k+2}, t_{k+3}\}$ is also decomposed in three disjoints sets, each of them constituting a stochastic clock:

- $\Gamma_1 = \{t_{k+1}\}$, $\Gamma_2 = \{t_k, t_{k+2}\}$, $\Gamma_3 = \{t_{k+3}\}$
- $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$
- $\Gamma_1 \cap \Gamma_2 \cap \Gamma_3 = \Phi$

This shows that the definition of a set C of observation classes decomposes a given sequence ω of timed observations $o(t_k)$ in a *superposition* of sequences ω_i, each of them being associated with a particular observation classe C_i:

- $C_1 = \{(x_i, low\} \Rightarrow \theta_1(C_1), \omega_1, \Gamma_1$
- $C_2 = \{(x_i, normal\} \Rightarrow \theta_{23}(C_2), \omega_2, \Gamma_2$
- $C_3 = \{(x_i, high\} \Rightarrow \theta_3(C_3), \omega_3, \Gamma_3$

The notion of observation class facilitates then the interpretation and the filtering of a given sequence ω whatever is the program $\Theta(X, \Delta)$. The next section shows that this notion provides also a powerful tool to model an observed process $(X(t), \Theta(X, \Delta))$.

3.1.5 Superposition Theorem

The program $\theta_{23}(\{x_i\}, \{normal\})$ of the latter example doesn't really exists: it is only a point of view, a way to read a sequence. This is why such a program is considered as an *abstract* program and is called an *abstract observer* in [?].

The concept of abstract observer is the core of the TOT. It is linked with the following *Superposition Theorem*:

Theorem 3.3 *Superposition Theorem*
If a program $\Theta(X, \Delta)$ is memoryless and the constants δ_i of Δ are independent, then any partition $\cup_{i=1\ldots n} \Delta_i$ of Δ so that $\forall i \neq j$, $\Delta_i \cap \Delta_j = \{\phi\}$ decomposes the program $\Theta(X, \Delta)$ in a superposition of n independent programs $\Theta_i(X_i, \Delta_i)$ so that:

1. $(\Delta = \bigcup\limits_{i=1\ldots n} \Delta_i) \Rightarrow X = \bigcup\limits_{i=1\ldots n} X_i$
2. $((\Delta = \bigcup\limits_{i=1\ldots n} \Delta_i) \wedge (X = \bigcup\limits_{i=1\ldots n} X_i)) \Rightarrow \Theta(X, \Delta) = \bigcup\limits_{i=1\ldots n} \Theta_i(X_i, \Delta_i)$
3. $(\Theta(X, \Delta) = \bigcup\limits_{i=1\ldots n} \Theta_i(X_i, \Delta_i)\}) \Rightarrow (X(t), \Theta(X, \Delta)) = \bigcup\limits_{i=1\ldots n} (X_i(t), \theta_i(X_i, \Delta_i))$

In other words, the partitioning of the set Δ in n disjoint sets Δ_i transforms a program $\Theta(X, \Delta)$ in a superposition of n independent programs $\Theta_i(X_i, \Delta_i)$. Such a partition is made with the definition of a set $C = \{C_i\}_{i=1\ldots n}$ so that:

$$\Theta(X, \Delta) = \bigcup_{i=1\ldots n} \theta_i(C_i) \tag{3.11}$$

It is important to note that the theorem 3.3 is only based on an *adequate partition* of the set Δ of constants and concerns only the program Θ: no hypothesis is made about the dynamic of the process $X(t)$.

The equation 3.11 and the theorem 3.3 means that, given an adequate set $C = \{C_i\}_{i=1\ldots n}$, any observed process $(X(t), \Theta(X, \Delta))$ can be seen as a *network* of observed processes $\bigcup\limits_i (X_i(t), \theta_i(C_i))$.

The term *network* is used because the partitioning of Δ does not entails the partition of the process $X(t)$: it does not matter that the subsets X_i are or aren't disjoints. In other words, the observed processes $(X_i(t), \theta_i(C_i))$ can *share* some time function $x_i(t)$. We will see in the next chapter that two observed process $(X_i(t), \theta_i(C_i))$ and $(X_j(t), \theta_j(C_j))$ are *connected* together when $X_i(t) \cap X_j(t) \neq \Phi$.

The application of these theorems is very simple. For example, considering collectively the theorems 3.2 and 3.3, each observation class C_i can be defined as a singleton so that each δ_i is associated with one and only one variable x_i (a variable name can be associated with multiple constants). In that case, a program $\Theta(X, \Delta)$ where Δ contains n constants δ_i can be considered as a superposition of n independent memoryless programs $\theta_i(C_i)$ (cf. equation 3.11. This way of defining the set C is very current in practice. And when the variable x_i is not known, an abstract variable ϕ_i can be used.

More generally, the importance of the Superposition Theorem comes from the fact it allows to describe recursively any complex observed process $(X(t), \Theta(X, \Delta))$ as a network of observed processes $\bigcup\limits_i (X_i, \theta_i(X_i, \Delta_i))$. This property is very important to diagnose a complex observed process $(X(t), \Theta(X, \Delta))$.

3.1.6 Temporal Binary Relation

The *Induction Theorem* is another important consequence of the Superposition Theorem [**?**]:

Theorem 3.4 *Relation*
Let $X(t) = \{x_i(t)\}_{i=1\ldots m}$ be a dynamic process where the values of two time functions $x_i(t)$ and $x_j(t)$ of $X(t)$ are linked together; let $\Theta(X, \Delta) = \bigcup\limits_{i=1\ldots n} \theta_i(C_i)$, $n \geq m$, be a program satisfying the Superposition Theorem 3.3 where $C = \{C_i\}_{i=1\ldots n}$ is the corresponding set of n observation classes satisfying the Observation Class theorem 3.2; let $C_i = \{x_i, \delta_i\}$ and $C_j = \{x_j, \delta_j\}$ two observation classes of C; let Γ_i and Γ_j be the stochastic clocks associated to the observation classes C_i and C_j.

If $\Theta(X, \Delta)$ contains two programs $\theta_i(C_i)$ and $\theta_j(C_j)$ so that the relation between the functions $x_i(t)$ and $x_j(t)$ implies a relation between the stochastic clocks Γ_i and Γ_j, then there exists a relation between the observation classes C_i and C_j.

This theorem looks like a tautology but it is very important: it means that if two time functions $x_i(t)$ and $x_j(t)$ are linked together, two programs $\theta_i(C_i)$ and $\theta_j(C_j)$ can be designed *to reveal* the existence of this relation. The condition is that the stochastic clocks Γ_i and Γ_j containing thetime stamps of the occurrences $C_i(t_{k_i})$ and $C_j(t_{k_j})$ are be linked together by a *temporal binary relation*:

Definition 3.4 *Temporal Binary Relation*
A temporal binary relation $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$, $\tau_{ij}^- \in \Re$, $\tau_{ij}^+ \in \Re$, is an oriented (sequential) relation between two observation classes C_i and C_j that is timed constrained with the $[\tau_{ij}^-, \tau_{ij}^+]$ interval.

The temporal constraint $[\tau_{ij}^-, \tau_{ij}^+]$ of a temporal binary relation $r(C_i, C_j, [\tau^-, \tau^+])$ is the time interval for observing an occurrence $C_j(t_{k_j})$ of the "output" observation class C_j after the observation of an occurrence $C_i(t_{k_i})$ of the "input" observation class C_i:

Definition 3.5 *Observed Relation*
Let the couple $(X(t), \Theta(X, \Delta))$ be an observed process defining a particular set $C = \{C_i\}$ of m observation classes containing two classes C_i and C_j; let $\omega = \{..., C_l(t_k), ...\}$, $t_k \in \Gamma \subseteq \Re$, $k = 0...n-1$, $l = 0...m-1$, be a sequence of n timed observations $C_l(t_k)$ provided by $(X(t), \Theta(X, \Delta))$.

A temporal binary relation $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ between two classes C_i and C_j is said to be observed in ω if there is at least two timed observations $C_i(t_{k_i})$ and $C_j(t_{k_j})$ so that $t_{k_j} - t_{k_i}$ satisfies the timed constraint $[\tau_{ij}^-, \tau_{ij}^+]$ of $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$.

Formally, the relation $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ is observed if and only if:

$$r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+]) \Leftrightarrow \exists C_i(t_{k_i}) \in \omega, \exists C_j(t_{k_j}) \in \omega,\ t_{k_j} - t_{k_i} \in [\tau_{ij}^-, \tau_{ij}^+] \tag{3.12}$$

A temporal binary relation of the form $r(C_i, C_j,]0, +\infty[)$ is a purely sequential binary relation: to be observed, the occurrence $C_j(t_{k_j})$ must succeed the occurrence $C_i(t_{k_i})$ (i.e. $t_{k_j} > t_{k_i}$). For simplicity reasons, such a sequential binary relation is denoted $r(C_i, C_j)$:

$$r(C_i, C_j,]0, +\infty[) \equiv r(C_i, C_j) \tag{3.13}$$

Now, all the elements are in place to provide a formulation of the fundamental *Theorem of Induction* of the TOT:

Theorem 3.5 *Induction*
Let $X(t)$ be a dynamic process containing two time functions $x_i(t)$ and $x_j(t)$; let $\Theta(X, \Delta))$ be a program associating two variable names x_i and x_j to the time functions $x_i(t)$ and $x_j(t)$ respectively; let C be a set of m observation classes build on $X \times Delta$; let C_i and C_j two observation classes of C that are respectively concerned with one and only one variable name x_i and x_j; let ω be a sequence of timed observations provided by the observed process $(X(t), \Theta(X, \Delta))$; let $\omega_i = \{C_i(t_{k_i})\}$, $k_i = 0...n_i - 1$, and $\omega_j = \{C_j(t_{k_j})\}$, $k_j = 0...n_j - 1$, be respectively two sequences of n_i and n_j timed observations of the classes C_i and C_j so that: $\omega_i \subset \omega$, $\omega_j \subset \omega$ and $\omega_i \cap \omega_j = \Phi$; let Γ_i and Γ_j be the two stochastic clocks defined by ω_i and ω_j.

1. *The existence of a binary relation between two stochastic clocks Γ_i and Γ_j induces the existence of a binary temporal relations of the form $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$.*
2. *A binary temporal relations of the form $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ subsumes the existence of a relation between two time functions $x_i(t)$ and $x_j(t)$ respectively.*

The Induction Theorem means that when two stochastic clocks are linked together, given that the corresponding observation classes are adequate according to the theorem 3.3, it is then justifiable to make the hypothesis of the existence of a relation between both functions that are at the origin of the clocks.

Obviously, the *confidence* in the existence of an induced relation between two time functions $x_i(t)$ and $x_j(t)$ depends on the *representativity* of the binary temporal relations $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$, and this latter is linked with the ratio of the number of observed relation in ω with the total number of pairs $(C_i(t_{k_i}), C_j(t_{k_j}))$, $t_{k_i} \leq t_{k_j}$, ω allows to build. This explains the introduction of the notion of probabilities in the TOT mathematical framework.

This theorem is the fundamental basis of the a new Knowledge Discovery in Database process called TOM4L (Timed Observation Mining for Learning) the presentation of which is out of the scope of this chapter. The interested Reader is invited to refer to [?, ?, ALG10, ?] for a description of TOM4L.

But this theorem is also fundamental to model an observed process $(X(t), \Theta(X, \Delta))$ because it imposes a *coherence* constraint on the model. The contraposee of this theorem says that if there is a link between the values of two timed function $x_i(t)$ and $x_j(t)$, then there *must be*:

1. a relation between the values of the corresponding variable name x_i and x_j,
2. a relation $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ between two adequate observation classes C_i and C_i, and
3. a relation between the stochastic clocks Γ_i and Γ_j.

Obviously, the abstract observers $\theta_i(C_i)$ and $\theta_j(C_j)$*must be* adequate according to the Superposition Theorem 3.3.

3.1.7 Abstract Chronicle Model

Nevertheless, the definition of a temporal binary relation $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ (cf. definition 3.4) is the basis of the notion of *Abstract Chronicle Model*:

Definition 3.6 *Abstract Chronicle Model*
Any arbitrarily made set $M = \{r_k(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])\}_{k=1\ldots n}$ *of* n *temporal binary relations* $r_k(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ *is an abstract chronicle model.*

The abstract chronicle models of the TOT framework are represented with a graphical knowledge representation language called "ELP" for "Event Language for Process behavior modeling" [?].

A particular sequence $\omega_i = \{o(t_k\}_{k=0\ldots n-1}$ of n timed observations that is consistent with the logical and the timed constraints of a given abstract chronicle model M is called an *instance* of M. For example, let us consider the following abstract chronicle model $M_{123} = \{\ r_{12}(C_1, C_2, [0,5]), r_{23}(C_2, C_3, [3,8])\}$. The sequence $\omega_i = \{C_1(1), C_4(3), C_2(4), C_1(8), C_3(10)\}$ contains the occurrences $C_1(1)$, $C_2(4)$ and $C_3(10)$ satisfying the logical and the timed constraints of M_{123}:

- $C_1(1)$ and $C_2(4)$ satisfy the logical condition of the relation $r_{12}(C_1, C_2, [0,5])$ (i.e. the observation class of $C_1(1)$ is C_1 (resp. C_2 for $C_2(4)$).
- $C_1(1)$ and $C_2(4)$ satisfy the temporal condition of the relation $r_{12}(C_1, C_2, [0,5])$ (i.e. $4-1 = 3$, $3 \in [0,5]$).
- $C_2(4)$ and $C_3(10)$ satisfy the logical condition of the relation $r_{23}(C_2, C_3, [3,8])$ (i.e. the observation class of $C_2(4)$ is C_2 (resp. C_3 for $C_3(10)$).
- $C_2(4)$ and $C_3(10)$ satisfy the temporal condition of the relation $r_{23}(C_2, C_3, [3,8])$ (i.e. $10-4 = 6$, $6 \in [3,8]$).

The abstract chronicle model and instance of model notions are of the most interest for the diagnosis of an observed process $(X(t), \Theta(X, \Delta))$. Specifically, a particular type of abstract chronicle models, called a *path*, play an important role:

Definition 3.7 *Path*
An abstract chronicle model made with a suite of $n-1$ *timed binary relations* $\{r_1(C_i, C_{i+1}, [\tau_1^-, \tau_1^+]), r_2(C_{i+1}, Ci+$ $1, [\tau_2^-, \tau_2^+]), ..., r_n(C_{n-1}, C_n, [\tau_n^-, \tau_n^+])\}$ *is a path.*

The $M_1 23$ abstract chronicle model, for example, is a path.

More generally, a set $P = \{..., p_i, ...$ of n_P pathes $p_i = \{..., r_{ni}^i(C_{k-1}, C_k, [\tau_k^-, \tau_k^+]))$ where the last relation $r_{ni}^i(C_{k-1}, C_k, [\tau_k^-, \tau_k^+]))$ of each path p_i, except one, is a relation contained in another path p_j of P is also called a path because in that case, P constitutes a kind of *path of paths*. Graphically represented, P is chained list of classes (cf. figure **??** for example, where the classes are the ellipses denoted with a number).

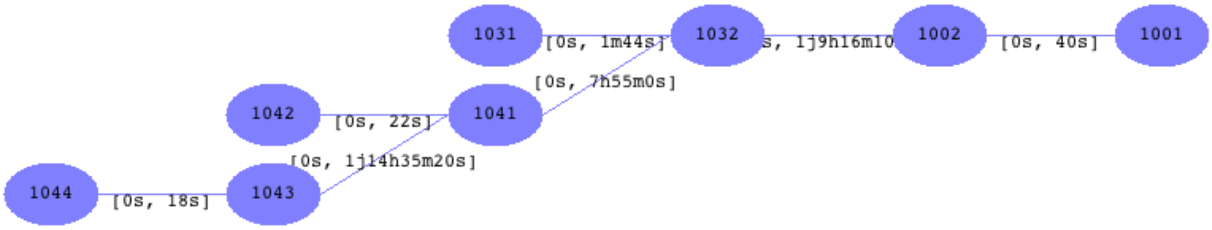


Figure 3.2: Example of a path made with 3 paths

A particular (set of) path P constitutes a specific structure that allows the "reading" of a given a sequence ω_i of timed observation $o(t_{k_i})$: if ω_i satisfies the logical and the temporal constraints of the suite of relations $r_j(C_{j-1}, C_j, [\tau_j^-, \tau_j^+])$, $\forall j$, of a given path P_i, then ω_i is an instance of the path P_i. In that case, the path P_i constitutes an *interpretation structure* of the sequence ω_i. This interpretation structure is like a *synopsis* of the *narative structure* that is the sequence ω_i. Inversely, ω_i is like a *scenario* (or the *story*) that *must* satisfy the logical and the temporal constraints of the path P_i.

3.2 Modeling with the TOT

Technically, the timed observations $o(t_{k_i})$ of a sequence ω_i constitute an *information flow*. A path P_i is a *representation* of the knowledge that is necessary to *interpret* the timed observations $o(t_{k_i})$ of ω_i.

Generally speaking, knowledge results of the interaction between an information flow and an arbitrary purpose. This interaction is assumed by humans which defines their purpose according to their own expectations [Non94], [Non91] and [AL01]. Information comes from all the possible sources: believes, observations, experimentation, scientific axioms, sensors, etc [Pol66], [NK98] and [SBF98]. The interaction is basically an interpretation of the information flow that traverses a thinking human [Dam05] and [Dam99].

To define the modeling principles of the TOT, the following operational definition of *knowledge* will be used:

Definition 3.8 *Operational Notion of Knowledge*
Knowledge results from an intentional interpretation of a flow of information.

This definition establishes a relation between knowledge, information and a purpose (an intention). The purpose is always defined by humans: in the framework of the TOT, the purpose is implemented in an observer program $\Theta(X, \Delta)$ which can be either executed by a human or a computer. Considering the diagnosis task of a dynamic process, the purpose is typically the assessment of a fault linked with the occurrence of an undesired behavior.

3.2.1 Model according to the TOT

The fundamental role of a model is the sharing of knowledge between humans. This sharing is facilitated through the mediation of signs belonging to a particular set (often called alphabet). These signs have no meaning in themselves but are necessary to represent knowledge in order to share a common understanding of an observed set of phenomenon.

As a consequence, a model is made with a particular arrangement of signs: the meaning results precisely of the specific arrangement the modeler choose to share its knowledge. The representation of a knowledge corpus requires then a set of rules that defines the authorized arrangements (i.e. a grammar).

This leads to define the notion of model according to the TOT:

Definition 3.9 *Model according to the TOT*
A model is an organized set of knowledge representations.

It is clear that the organization of the knowledge representations within a model is of the main importance. The Timed Observations Theory being concerned with the evolutions of a process over time, the knowledge under consideration is linked with the relations between functions of time $x_i(t)$, the constants δ_i and the stochastic clocks Γ_i=$\{t_{k_i}\}$, t_{k_i}, $k_i \in N$. The TOT organizes these relations around the notion of variable x_i (cf. figure 3.3).

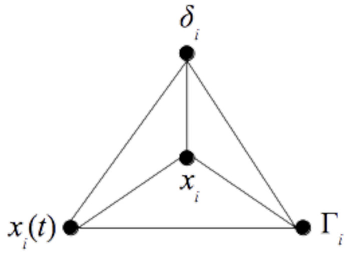


Figure 3.3: Relations between the Basic Objects of the TOT

A piece of knowledge belongs then to three fundamental categories:

1. Relations between the functions $x_i(t)$ of a process $X(t)$. This category of knowledge is called the "structural knowledge" because, in the framework of the TOT, the functions $x_i(t)$ are the constituents (i.e. the components) of a process $X(t)$.

2. Relations between the constants δ_i of the set Δ used by a program Θ to describe the evolutions of the functions $x_i(t)$. This category is called the "functional knowledge" because the relations between the constants δ_i can be represented with logical rules linking together subsets Δ_i of constants of Δ and so, specifies abstract mathematical functions under the form of "tables of values".

3. Relations between the stochastic clocks Γ_i. This category is called the "behavioral knowledge" because these relations describes the links between the evolutions of the functions $x_i(t)$ of a process $X(t)$. This type of knowledge is directly linked with the set $C = \{C_i\}_{i=1..n_c}$ of observation classes that will be define to the modeling.

As figure 3.3 aims at showing, these three categories of relations are linked together: a specific set Δ of constant δ_i will lead an observer program Θ to describe the evolution of a process $X(t)$ with a particular set Γ of time stamps. The role of the concept of "variable" in the Timed Observations Theory framework is to provide the mean to analyze the consistency of these three categories of knowledge about a process $X(t)$.

So, the concept of variable defines a supplementary category of knowledge, which is a kind on "meta-knowledge", that fundamentally defines the way a dynamic process$X(t)$ is perceived by humans (i.e. the modeling point of view). In practice, this leads to the following definition of the aim of the modeling activity:

Definition 3.10 *Modeling activity according to the TOT*
The modeling activity of a dynamic process $X(t)$ aims at representing the elicited knowledge according to a formalism and at distributing the knowledge representations over three models, the structural, the functional and the behavioral model, according to a definition of a particular set of variables X.

By construction, a particular set of variables X is a subset of all the variables that can be defined about a dynamic process. The only rational way to specify X is precisely the modeling purpose: are only required the variables that play a role in the modeling purpose (i.e. the diagnosis task in this case). The other variables can be forgotten (at least in a first step). This set of variables X defines then the process according to modeling purpose, and so fixes the abstraction level of the model.

Finally, a modeling methodology must defines the organization laws and the representation rules of knowledge (i.e. the representation formalism). Within the TOT framework, the formalism must allow the expression of relations between the main concept of the TOT.

3.2.2 Abstraction Level

The aim of a modeling framework is to provide the tools allowing the building of a model that:

1. resides at any level of abstraction,

2. is logically coherent (i.e. contains no contradiction), and

3. is as complete as possible.

These goals are given in the order of their importance: clearly, providing a coherent model at the right abstraction level is the main modeling law of TOM4D, its completeness being desired but does not constitute a primary condition.

The experts' knowledge can be view as a set of propositions that the experts formulate according to this model. In other words, any proposition formulated by an expert is, by hypothesis, an assertion about a property of the process. These propositions are concerned with the structure, the functions, the behavior or the role the process play in a exploitation purpose.

Within the mathematical framework of the TOT, the abstraction level is defined by the observed process $(X(t), \Theta(X, \Delta)$ it self: no constraint is made about the way the sets $X(t)$ and Δ are made (The TOT imposes only two constraints on the *program* Θ). Consequently, to use the mathematical framework of the TOT, an analysis of the available knowledge must be made in order to define the sets $X(t)$ and Δ. We will see in the next chapter that the TOM4D methodology resorts on the combinaison of the knowledge interpretation framework of the CommonKADS methodology (cf. [SAA^{+}00] and [BdV94]) with the Tetrahedron of States (ToS) [RK83] to interpret the available knowledge and to analyze its soundness.

But, whatever is the interpretation framework, the definition of a particular set X of variables constitutes the *core* of the modeling process. So, to assess the *coherence* of the model, the six types of binary relations of figure 3.3 *must* be examined. This leads to distribute the knowledge representations over the three basic models of the TOT: the structural, the functional and the behavioral models. This allows to use Reiter's Logical Theory of Diagnosis [Rei87, **?**] (cf. [**?**], [**?**] and [**?**] for an example), that is to say the Formal Logic (i.e. in particular the predicate calculus) to analyse the internal coherence of a model.

The completeness of a model of a dynamic process corresponds to the property of such a model to allow to provide any proposition an expert can formulate about the corresponding process. So to assess the completeness of a model, it necessary but sufficient to check first the completeness of the variable set X and next, to verify that all the possible binary relation between two variables x_i and x_j of X has been examined according to the three fundamental dimensions of knowledge that are the structural, the functional and the behavioral dimensions.

As a consequence, the TOT mathematical framework invites to combine a syntactic and a semantic approach of a modeling process and provides tools to control the knowledge acquisition process and identify the main modeling concepts of the dynamic process (variables, constants, values, thresholds, components, states, etc).

3.2.3 Modeling Principles

Clearly, the notion of *variable* according to the TOT provides five fundamental modeling principles:

1. **Variable localization.**
   A time function $x_i(t)$ is a *signal* provided by a *sensor* located at a particular place defined as a *component*. So, a function $x_i(t)$ specifies a variable $x_i(t)$, a component c_i and a binary relation that associate $x_i(t)$ to c_i. As a consequence, a variable $x_i(t)$ is always associated with a sensor that is physically located on a component c_i. In other words, any variable $x_i(t)$ of X must be associated with one and only one component c_i.

2. **Multi-value variable.**
   A variable $x_i(t)$ is necessarily defined over a set Δ_{x_i} of possible values containing at least two elements. This means that when the experts' knowledge defines only one value δ_i for a variable $x_i(t)$, the knowledge engineer must introduce in Δ_{x_i} another constant, denoted δ_j for example, meaning "not δ_i" (i.e. $\Delta_{x_i} = \{\delta_i, \delta_j\}$ and $\delta_j \equiv \neg\, \delta_i$). This principle is a direct consequence of the spatial segmentation of the Timed Observation Theory (cf. Figure 3.3).

3. **Discernible state.**
   According to the interpretation 3.8, an occurence $C_i(t_k)$ of an observation class C_i corresponds to the assignment of a value δ_i to a variable x_i. Such an assignation results necessarily of an *observable* modification in the dynamic process $X(t)$. So two occurrences $C_i(t_k)$ and $C_j(t_{k+1})$ marks a *observable state transition* in an observed process $(X(t), \Theta(X, \Delta)$. This means that a temporal binary relation $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ defines a particular *discernible state.*

4. **Knowledge of different nature separation.**
   Since the TOT defines four categories of knowledge (structural, functional, behavioral and perception), four models will contains a specific category of knowledge representations: a *Structural Model SM* will contain all the structural knowledge, a *Functional Model FM* will contain all the functional knowledge, a *Behavioral Model BM* will contain all the behavioral knowledge, and a *Perception Model PM* will contain the perception knowledge. This constitutes the multi-modeling framework of the TOT [**?**], [CR99] and [ZGF06].

5. **Symbol driven modeling.**
   The knowledge interpretation aims at identifying the minimal set of symbols denoting a time function $x_i(t)$, a variable x_i or a constant δ_i and the minimal set of relations between them (cf. figure 3.3). The logical properties coming from these minimal sets are necessary and sufficient to complete the model. Among other meaning, this principle means that the introduction of a symbol that is not associated with an element of the domain knowledge is prohibited.

The *Discernible state* principle is particularly important according to the notion of *Behavioral Model BM.* A *discernible state* is a property of the *model* of an observed process$(X(t), \Theta(X, \Delta))$:

Definition 3.11 *Discernible State*
An arbitrary made set $\{r_k(C_{i_k}, C_{j_k})\}_{k=1\ldots n_k}$, $\forall k, i_k \neq \jmath_k$, of n sequential binary relations $r_k(C_{i_k}, C_{j_k})$ is a discernible state.

This notion is then conceptually different of the classical *state* notion of the Discrete Event System (DES) domain where *a state represents a property of the process $X(t)$* itself.

3.3 Conclusion

Generally speaking, fault assessment is a knowledge intensive task that requires a model of the process under consideration.

According to the Timed Observations Theory, the notion of fault is concerned with particular behaviors of a dynamic process: a fault is linked with the occurrence of an undesired behavior of an observed process $(X(t)$, $\Theta(X, \Delta))$. This means that the required knowledge to assess a fault about a dynamic process is the one required by the tasks of monitoring, diagnosis and prognosis:

- the monitoring task requires the knowledge to infer the current behavior and to categorize it as desirable or undesirable behavior,
- the diagnosis task requires the knowledge to infer the causes of the undesirable behaviors,

- and the prognosis tasks requires the knowledge to infer the potential future undesirable behaviors that can result from the current situation described by the monitoring and the diagnosis tasks.

The quality of the knowledge corpus required by these tasks is directly linked with the pertinence of the fault assessment task. If this point is quite trivial, the knowledge engineering method used to acquire and represent this knowledge corpus must provide the tools to guarantee an adequate level of quality.

This chapter introduces the mathematical tools of the Timed Observation Theory that are required to define a multi-model framework and the laws to distributes the knowledge representation on the right model. Five modeling principles have been derived from these mathematical tools. These five principles constitutes a strong logical basis for the modeling work of the knowledge engineer: from the identification of the variables, the knowledge engineer will identify the possible values a variable can take over time, its corresponding function and observation classes and so, it defines the discernible state space of an observed dynamic process. Next, the knowledge analysis examines all the possible and the impossible relations between two elements (variable, constant, component and observation classes), conducted through their semantic properties. The organization of the resulting knowledge representations in the four models leads to an operational model of the dynamic system.

The next chapter presents the TOM4D methodology (Timed Observation Methodology for Diagnosis) that implements the TOT modeling framework.

CHAPTER 4

The Timed Observation Modeling for Diagnosis Methodology

TOM4D (Timed Observations Modeling for Diagnosis) is a knowledge engineering modeling approach for dynamic systems focused on timed observations. The aim of TOM4D is to produce a model of an *observed process* $(X(t), \Theta(X, \Delta)$ that is suitable for knowledge intensive tasks about dynamic process as for example supervision, diagnosis, prognostic and control tasks.

One the main specificities of the TOM4D methodology is to be a timed observation-centered modeling approach designed to be applied to model dynamic processes. The inputs of a TOM4D modeling process are both timed observations provided by a database and experts' a priori knowledge.

A TOM4D model presents the following major properties:

1. It can be faced with real world timed data (i.e. sequences of timed data making scenarios).
2. Any experts' assertion about the dynamic process can be faced to with TOM4D model.

These two major properties are fundamental when reasoning about a dynamic process: the first one allows to assess the adequateness of a model with real world data and the second allows the check the experts' knowledge with an already validated model. Furthermore, the second property constitutes a way to analyze the completeness and the coherence of a model under construction. These two properties directly comes from the modeling principles of the TOT described in the precedent chapter.

After a brief introduction presenting the aim of TOM4, this chapter presents the formalisms defined to represent the TOM4D's models and next, describes the tools that are used to control the introduction of the semantic in the global modeling process. These elements are then applied on the didactic example of [CPR00] that is used to illustrate the presentation of the TOM4D methodology.

4.1 Introduction to the TOM4D Methodology

The conceptual modeling framework of TOM4D distributes the available knowledge representations about an observed dynamic process $(X(t), \Theta(X, \Delta))$ over four models [Zan04, LGMC08, LGM07]):

- a *Structural Model SM* describing the relations between the time function $x_i(t)$ of a process $X(t)$,

- a *Functional Model* FM providing the relations between the values δ_i of the variables x_i,
- a *Behaviour Model* BM defining the observation classes C_i triggering the discernible state transitions within the observed dynamic process $(X(t), \Theta(X, \Delta)$, and
- a *Perception Model* PM defining the way the observed dynamic process $(X(t), \Theta(X, \Delta)$ is seen by the model.

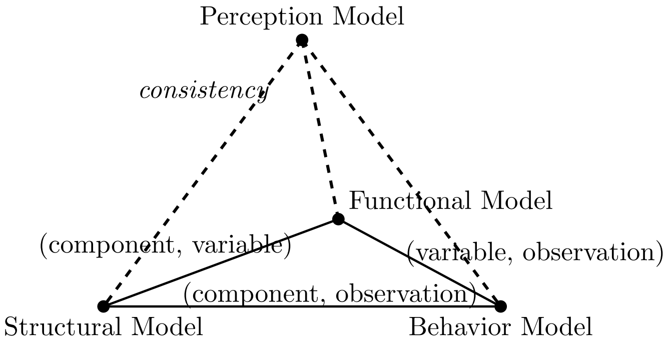


Figure 4.1: Relations between the TOM4D models

Figure 4.1 illustrates the modeling framework of an observed process $(X(t), \Theta(X, \Delta)$ according to the the TOT. This conceptual framework is centered on the TOT notion of variable: the structural model SM associates any variables used in a function of the functional model FM with a component (or a component aggregate) and a timed observation class of the behavioral model BM associates a variable to a constant. So, a timed observation being an occurrence of an observation class, it corresponds to the assignment of a constant to a variable that play a role in at least one function of the functional model FM and is located with one of the components of the structural model SM. The perception model PM defines the minimal set of variables and the corresponding minimal set of constants that are required to define the goals of a dynamic process. Doing so, the perception model specifies a minimal set of constraints that the functional, structural and behavioral models must respects. This means that any relation defined in these models must be consistent with the constraints of the perception model PM.

The aim of the TOM4D methodology is to allow the building of a model that resides at the same abstraction level as the experts' knowledge. To assess the abstraction level of the model, TOM4D resorts to a knowledge interpretation framework in order to introduce, in the modeling process, the semantic content provided by experts in a gradual and controlled way. This knowledge interpretation framework is based on the conceptual models of the CommonKADS [SAA$^+$00, BdV94] methodology to interpret the available experts' knowledge, and uses the Tetrahedron of States (ToS) [RK83] to analyze its soundness. This interpretation of the available knowledge leads to define a particular set X of variables constituting the core of the process model so that the six types of binary relations of figure 4.1 can, and must, be examined.

To assess the coherence of the model, the knowledge representations are distributed over the three basic models of the TOT, the structural, the functional and the behavioral models, according to the meaning of the corresponding knowledge. TOM4D has then been design as a primarily syntax-driven approach [PL10, LGMC08, LGM07] so that Reiter's Logical Theory of Diagnosis [Rei87] can be used to assess the coherence of the model, and in return, to supply tools to assess the experts' knowledge [Dag01]. To this aim, the knowledge representation formalism is based on the Formal Logic (i.e. the predicate calculus).

Keeping in mind that a TOM4D model aims at being the experts' model of a dynamic process, the experts' knowledge can be view as a set of propositions that the experts formulate according to this model. In other words, any proposition formulated by an expert is, by hypothesis, an assertion about a property of the process. These propositions are concerned with the structure, the functions, the behavior or the role the process play in a exploitation purpose. This means that the completeness of a TOM4D model corresponds to the property of such a model to allow to provide any proposition an expert can formulate about the corresponding process. So to assess the completeness of a TOM4D model, it necessary but sufficient to check first the completeness of the variable set X and next, to verify that all the possible binary relation between two variables x_i and x_j of X has been examined according to the three fundamental dimensions of knowledge that are the structural, the functional and the behavioral dimensions. Again, the syntactic based modeling approach of TOM4D facilitates the analysis of the completeness of a model.

As a consequence, the combination of a syntactic and a semantic approach in the TOM4D modeling principles allows to provide tools for the knowledge engineers to control the knowledge acquisition process and identify the main modeling concepts of the dynamic process (variables, constants, values, thresholds, components, states, etc).

Before providing the details about the tools allowing a control over the semantic, let us examine the formalism of the four TOM4D models.

4.2 TOM4D Model Formalism

This section provides the TOM4D fomalisms that are defined to represents the differents models of a dynamic process, that is to say the *Perception Model*, the *Structural Model*, the *Functional Model*, the *Behavioral Model* and finally, the *Network Model* of a set of connected dynamic processes.

Each of them have a textual and a graphical representation, except the *Perception Model* that is only textual. So the textual definition of each model will be illustrated with a graphical representation that comes from the models of the didactic example of [CPR00]. Of course, the model of this example will be provides both textually and graphically.

In this section, let us consider a dynamic process $X(t) = \{x_i(t)\}_{i=1...n}$ of n time functions $x_i(t)$. To each time function x_i is associated a set $X = \{x_i\}_{i=1...n}$ of n variable names x_i. To each variable x_i of X is associated a *definition domain* $\Delta_{x_i} = \{\delta_j^{x_i}$ where Δ_{x_i} is a set of constants $\delta_j^{x_i}$ denoting the possible values a variable x_i can take.

4.2.1 Perception Model Formalism

The *Perception Model* $PM(X(t))$ defines the contraints of a process $X(t)$ in terms of its physical and informational borders and its operating modes:

Definition 4.1 *Perception Model Formalism*
A Perception Model $PM(X(t))$ of a dynamic process $X(t)$ is a structure $< X, \Psi, R^q >$ where:

- *X is the set of variable names x_i associated with the time function $x_i(t)$ of the process $X(t)$,*

- *Ψ is a finite set of constant values $\psi_i \in \Re$ corresponding to thresholds.*

- *$R^q = R^{goal} \cup R^n \cup R^{ab}$ is a set of predicates defining ranges of values with the elements of Ψ in order to determine the subset R^{goal} of the operating goals, the subset R^n of the normal operating modes and the subset R^{ab} of the abnormal operating modes as constraints on the values of the time functions $x_i(t)$ of $X(t)$.*

For example, the Perception Model $PM(X(t))$ of the hydraulic process $X(t) = \{x_1(t), x_2(t), x_5(t), x_6(t), x_7(t)\}$ is the structure $< X, \Psi, R^q >$ where:

- $X = \{x_1, x_2, x_5, x_6, x_7\}$,

- $\Psi = \{\Psi_{51}, \Psi_{52}, \Psi_{60}, \Psi_{71}, \Psi_{72}\}$

- $R^q = R^{goal} \cup R^n \cup R^{ab}$.

  - $R^{goal} = \{\exists t_0, \exists t_i, t_i \geq t_0, \forall t \geq t_i, x_7(t) \geq \Psi_{72}\}$
  - $R^n = \{\forall t \geq t_i, x_7(t) \geq \Psi_{72} \wedge \forall t \geq t_0, x_6(t) \leq \Psi_{60}\}$
  - $R^{ab} = \{\forall t \geq t_i, x_7(t) < \Psi_{72} \vee \forall t \geq t_0, x_6(t) > \Psi_{60}\}$

4.2.2 Structural Model Formalism

A TOM4D structural model $SM(X(t))$ specifies the components, denoted c_i, of a dynamic process $X(t)$, the interconnections between them and the link of each system variable x_i to a component c_i. Two components c_i and c_j can be related each other by the mean of input and output ports, respectively denoted $in(c_i)$ and $out(c_i)$. According to the modeling priciples of the TOT, all the variables x_i of X must be associated to one port of a component c_i.

Definition 4.2 *Structural Model Formalism*
A structural model $SM(X(t))$ of a dynamic process $X(t)$ is a structure $< COMPS, R^p, R^x >$ where,

- *$COMPS = c_{ii=1 \ldots n_c}$ is a finite set of constants c_i denoting the process components,*

- *R^p is a set of equality predicates of the form $out(c_i) = in(c_j)$ defining the interconnections between two components c_i and c_j of $COMPS$,*

- *R^x is a set of equality predicates of the form $out(c_i) = x_j$ linking a variable x_j of X to one and only one output port of a component c_i.*

The figure 4.2 presents a structural model made with 6 components. The definition 4.2 has been made so that a TOM4D structural model specifies as *"structural model"* in Reiter's logical theory of Diagnosis.

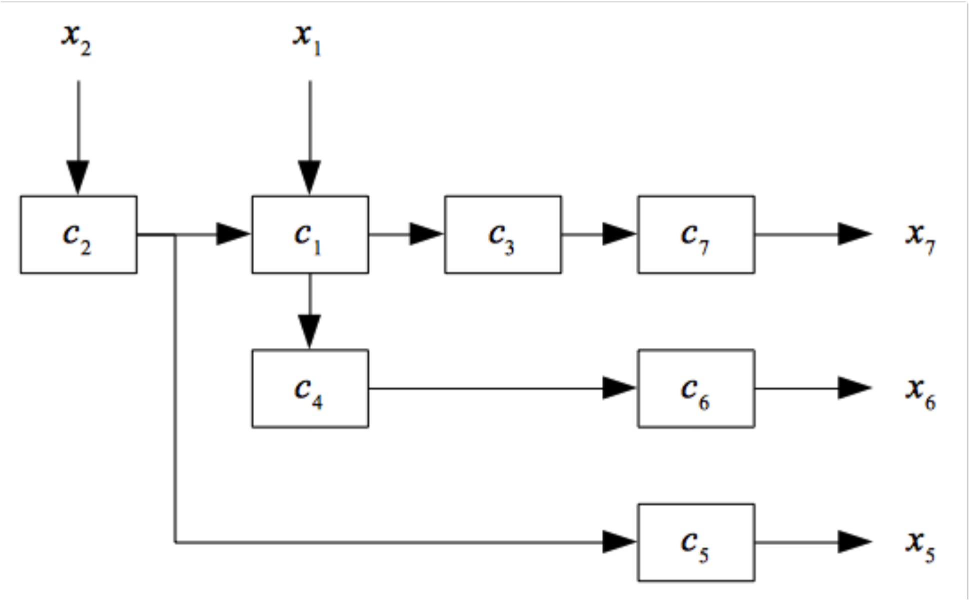


Figure 4.2: Graphical Representation of a Structural Model SM

$f_1 : \Delta_{x_2} \rightarrow \Delta_{x_5}$		
x_2	$f_1(x_2)$	*Interpretation:* $x_5 = f_1(x_2)$
δ_{20}	δ_{50}	$x_2 = \delta_{20} \Rightarrow x_5 = \delta_{50}$
δ_{21}	δ_{51}	$x_2 = \delta_{21} \Rightarrow x_5 = \delta_{51}$
δ_{22}	δ_{52}	$x_2 = \delta_{22} \Rightarrow x_5 = \delta_{52}$

Table 4.1: Example of the Algebraic Representation of a function $x_5 = f_1(x_2)$

4.2.3 Functional Model Formalism

A *Functional Model* $FM(X(t))$ aims at providing the complete set of logical relations between the values $\delta_j^{x_i}$ of the variables x_i of $X(t)$. The logical relations are provided using the rule formalism of the formal logic (i.e. implication logical connector "$\Rightarrow$"). They are described with *tables of values* (cf. table 4.1) corresponding to the algebraic representation of *mathematical functions* denoted $f_i()$.

Definition 4.3 *Functional Model Formalism*
A Functional Model $FM(X(t))$ of a dynamic process $X(t)$ is a structure $< \Delta, F, R^f >$ where,

- $\Delta = \bigcup_{x_i \in X} \Delta_{x_i}$ *is the union of all the sets of constant Δ_{x_i} defined for the variables x_i of X.*

- $F = \{f_i : \Delta^{x_1} \times ... \times \Delta^{x_k}\}$ *is a set of functions $f_i(x_1, x_2, ..., x_k)$ defined on $\Delta^{x_1} \times ... \times \Delta^{x_k}$ providing a particular constant $\delta_i^{x_k}$ for each combinaison $\delta_j^{x_1} \times ... \times \Delta_l^{x_{k-1}}$.*

- $R^f = \{x_r = f_i(x_1, ..., x_k)\}$ *is a set of equality predicates $x_r = f_i(x_1, ..., x_k)$ linking the values Δ_{x_r} of a variable x_r with a function f_i of F.*

The figure 4.3 graphically represents a functional model FM made with four functions $x_3 = f_1(x_2)$, $x_7 = f_2(x_3)$, $x_6 = f_3(x_3, \phi_8)$ and $x_1 = f_4(\phi_8)$ linking together the values of the six variables x_1,x_2, x_3, x_5, x_6, x_7 and ϕ_8.

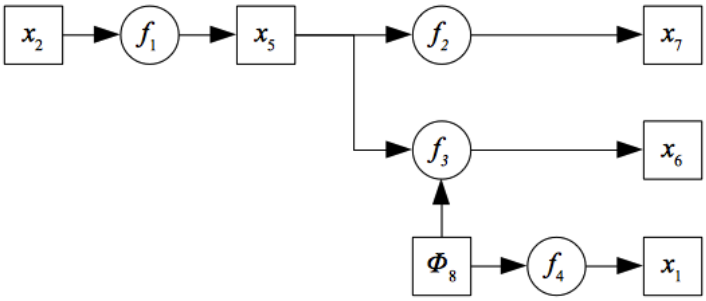


Figure 4.3: Graphical Representation of a Functional Model FM

A TOM4D functional model is then made with a set of algebraic representations of logical functions. Each functions specifies then a *knowledge base* containing a set of *production rule* of a particular form (cf. the CommonKADS notion of *model of rules*). Each rule corresponds to a particular *If... Then...* formula that an expert can verbalize about the logical consequence of the fact that a subset of variables takes some particular values on the value of another variable (cf. equation 4.1).

$$x_1 = x_{1i} \wedge x_2 = x_{2j} \wedge ... \wedge x_n = x_{nk} \Rightarrow y = y_i \tag{4.1}$$

The *production rule* form is adequate to be used with the *Modus Ponens deduction rule* that logicaly represents *If... Then...* propositions (cf. equation 4.2).

$$\begin{aligned} &\vdash x_1 = x_{1i}, x_2 = x_{2j}, ..., x_n = x_{nk}, \\ &\vdash x_1 = x_{1i} \wedge x_2 = x_{2j} \wedge ... \wedge x_n = x_{nk} \Rightarrow y = y_i \\ &\vdash y = y_i \end{aligned} \tag{4.2}$$

The aim of this representation is that a TOM4D functional model plays the same role as the *"generic component model behavior"* in Reiter's logical theory of Diagnosis. Technically, TOM4D simply uses the algebraic formalism (i.e. *table of values*) to represent sets of rules respecting the same model. Besides being compatible with Reiter's logical theory of diagnosis, the main advantages is the compactness of this representation and the fact that it is much more *natural* and *usual* to experts than production rules. These advantages are fundamentals to the validation of a model.

4.2.4 Behavioral Model Formalism

A TOM4D behavioral model $BM(X(t))$ aims at describing the possible sequences of observation classes that can occur; and therefore, the discernible states between them.

Definition 4.4 *Behavioral Model Formalism*
A behavioral model $BM(X(t))$ of a dynamic process $X(t)$ is a structure $< S, C, R^s >$ where:

- $S = \{s_k = \{r(C_i, C_j, [\tau^-_{ij}, \tau^+_{ij}])_{i \neq j}\}\}_{k=1...n_s}$ *is a set of n_s predicate equal linking a discernible state s_k with an arbitrary made set $\{r(C_i, C_j, [\tau^-_{ij}, \tau^+_{ij}])\}_{i \neq j}$ of timed binary relations $r(C_i, C_j, [\tau^-_{ij}, \tau^+_{ij}])$ between an observation classe C_i of C to another C_j where $i \neq j$.*
- $C = \{C_i\}_{i=1...n_c}$ *is a set of n_c observation classes $C_i = \{(x_i, \delta^{x_i}_j)\}_{j=1...n_{ci}}$ where each class C_i is made with one and only one variable x_i.*

- $R^s = \{s_j = \gamma(s_i, C_k)\}_{i\neq j}\}_{k=1\ldots n_{R^s}}$ *is an arbitrary made set of* n_{R^s} *predicates equal linking a discernible state* s_j *of* S *to the output of the* γ *modeling function defined on* $S \times C$ *which satisfies the two following constraints:*

  1. $\exists C_\phi \in C, r(C_\phi, C_k[\phi^-_{ij}, \phi^+_{ij}]) \in s_i$.
  2. $\exists C_\phi \in C, r(C_k, C_\phi, [\phi^-_{ij}, \phi^+_{ij}]) \in s_j$.

The set S provides then the *semantic* of the discernible states s_i of the process. Technically, according to this definition, a discernible state s_k being nothing more than a *set* of timed binary relations $r(C_i, C_j, [\tau^-_{ij}, \tau^+_{ij}])$, the *semantic* of a discernible states s_i is then defined by its set $\{r(C_i, C_j, [\tau^-_{ij}, \tau^+_{ij}])_{i\neq j}\}$ of timed binary relations. When the time constraints is unknown, a timed binary relations $r(C_i, C_j, [\phi^-_{ij}, \phi^+_{ij}])$ is simply written under the form of a sequential binary relation $r_{ij}(C_i, C_j)$. Such a sequential binary relations must be read $C_i \to C_j$.

The set R^s defines the relations between two discernible states s_i and s_j of S through a particular observation class C_i. The modeling function γ defines the relation between two discernible states: s_i and s_j are linked together by γ if and only if $\forall i \neq j, s_j = \gamma(s_i, C_k)\}$. γ is a modeling function because it specifies the possible paths (cf. definition 3.7) within a behavioral model. These possible paths will play a basic role in a diagnosis reasoning. According to the DES, the set R^s, that is to say the modeling function γ, specifies the *state transitions* within an automata. The only specificity of a TOM4D behavioral model is then that a *discernible state transition* between two *discernible states* s_i and s_j is conditioned with the occurrence of a particular observation class C_{m_i}.

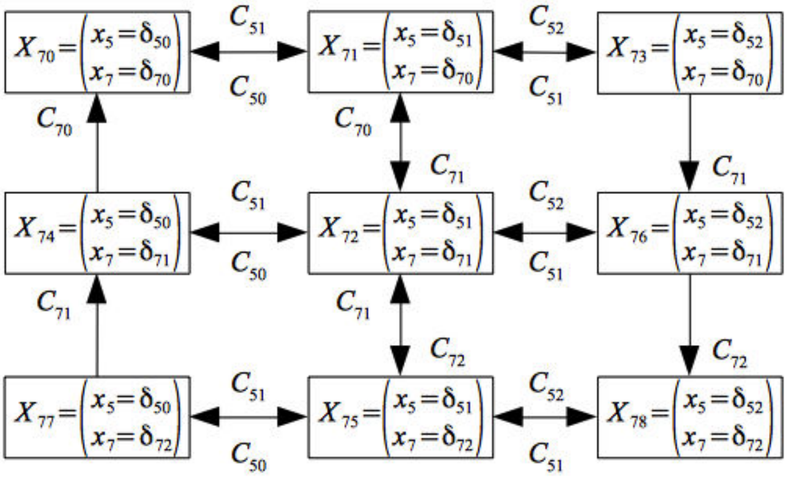

$X_{70}=\begin{pmatrix}x_5=\delta_{50}\\x_7=\delta_{70}\end{pmatrix}$
C_{51}
C_{50}
$X_{71}=\begin{pmatrix}x_5=\delta_{51}\\x_7=\delta_{70}\end{pmatrix}$
C_{52}
C_{51}
$X_{73}=\begin{pmatrix}x_5=\delta_{52}\\x_7=\delta_{70}\end{pmatrix}$
C_{70}
C_{70}
C_{71}
C_{71}
$X_{74}=\begin{pmatrix}x_5=\delta_{50}\\x_7=\delta_{71}\end{pmatrix}$
C_{51}
C_{50}
$X_{72}=\begin{pmatrix}x_5=\delta_{51}\\x_7=\delta_{71}\end{pmatrix}$
C_{52}
C_{51}
$X_{76}=\begin{pmatrix}x_5=\delta_{52}\\x_7=\delta_{71}\end{pmatrix}$
C_{71}
C_{71}
C_{72}
C_{72}
$X_{77}=\begin{pmatrix}x_5=\delta_{50}\\x_7=\delta_{72}\end{pmatrix}$
C_{51}
C_{50}
$X_{75}=\begin{pmatrix}x_5=\delta_{51}\\x_7=\delta_{72}\end{pmatrix}$
C_{52}
C_{51}
$X_{78}=\begin{pmatrix}x_5=\delta_{52}\\x_7=\delta_{72}\end{pmatrix}$

Figure 4.4: Graphical Representation of a Behavioral Model BM

To facilitate the reading of a behavioral model, it is usual to label the discernible states s_k with a vector, typically called the *discernible state vector* and denoted $\underline{X}$, of n dimensions corresponding to the number of variables x_i of the set X so that $\underline{X}$ is defined over the *discernible state space* $\Delta_{x_1} \times \ldots \times \Delta_{x_i} \ldots \times \Delta_{x_n}$. As a consequence, a particular value $\underline{X_k}$ of the vector $\underline{X}$ corresponds to the assignation of a particular value δ_{ij} to each coordinated i of $\underline{X}$ (i.e. $\forall i, x_i = \delta_{ij}$). This leads then to the bijective correspondence: $s_k \leftrightarrow \underline{X_k}$. When there is no confusion with the set X of variable names, the vector $\underline{X}$ (and its values $\underline{X_k}$) is simply denoted X (resp. X_k). The figure 4.4 provide a graphical representation of a behavioral model BM in which this type of label has been used. In this figure, the boxes represent the discernible states that have been labeled with the values X_{7i}, $i = 0...8$, of a two dimensions vector corresponding to two variables x_5 and x_6, so that each label X_{7i} corresponds to a particular state s_{7i}.

4.2.5 Relation between Functional and Behavioral Models

The algebraic representation of a function cannot directly represent a function $y(t) = f(x(t))$ containing an integral operator as in the equation 4.3.

$$y(t) = \int x(t)dt \tag{4.3}$$

Such functions require an adequate transformation in order to have an algebraic representation of such an equation. A usual transformation is the *Laplace* transformation in the Hilbert's mathematical space (i.e. the vectorial continuous space defined on $\Re$ where the exponential functions are the basis vectors). When the space is discrete, that is to say defined on N, the classical representation of such a function is the *Finite State Machine* (FSM) representation where the output variable $y(t) \in N$ is a function of the internal state q of the FSM ($q \in \{q_i\}_{i=0...n_q} \in N$) and the input variable $x(t) \in N$ (equation 4.4).

$$y(t) = f(x(t), q) \tag{4.4}$$

Technically, a function $x(t) = f(x(t), q)$ is made with two sub-functions defined with the equations 4.5. In these equations, $q(t_k)$ denotes the current value of q at time $t = t_k$ and $q(t_{k-1})$ its next value at time t_{k-1} if and only iff the condition made on the values of x are satisfied.

$$\begin{array}{rcl} q(t_k) & = & f_{x \to q}(x(t_k), q(t_{k-1})) \\ y(t_k) & = & f_{q \to y}(x(t_k), q(t_k)) \end{array} \tag{4.5}$$

The equations 4.5 shows that the internal variable q play the role of a memory: the variable q *stores* the last value q_i that has been entailed by the input value of $x(t)$. The following points have to be noticed about these equations:

- The functions $x(t)$ and $y(t)$ are piecewise constant functions of the continuous time $t \in \Re$.
- The FSM equations 4.5 manage the state transitions at some particular time stamps t_k: $t_k - t_{k-1} \in \Re$ can be as small as possible.
- In practice, such a FSM is implemented with a sequential machine, like a Turing's machine (i.e. a computer) for example. In that case, the time is discrete and then $t_k - t_{k-1} \in N$ is a function of the internal clock of the sequential machine.
- The values q_i have *no meaning* in them-self: it is only through the function $f_{q \to y}$ that a particular meaning *can be* attributed to the values q_i.

The differents internal states q_i, $i = 0...n_q$ of a FSM can be represented with a discrete variable, let us say ϕ, defined on a set Δ_ϕ of values $\{\delta_{\phi_i}\}_{i=0...n_q}$. The values δ_{ϕ_i} of Δ_ϕ enters in the value table describing the function $y(t) = f(x(t))$ of the equation 4.3 as *conditions* linking particular values x_i of $x_i(t)$ with particular values y_i of y (cf. equation 4.6).

$$\phi = \phi_i \wedge x = x_i \Rightarrow y = y_i \tag{4.6}$$

It is to note that the algebraic representation of function containing at least one integrator (cf. equation 4.3) is *always* possible: such a transformation is a classical result of the Automata Theory and is some times called the *time to space* transformation. For example, the table 4.2 provides the algebraic representation the function $f_3(x_5, \phi_8)$ of figure 4.3 containing one integrator. In this function, the variable ϕ_3 is *internal*: its role is to represents the effects of the different *states* in which $f_3(x_5, \phi_8)$ can be.

$f_3 : \Delta_{x_5} \times \Delta_{\phi_8} \rightarrow \Delta_{x_6}$			
ϕ_3	x_5	ϕ_8	$f_3(x_5, \phi_8)$
δ_{30}	-	δ_{81}	δ_{61}
δ_{30}	δ_{50}	-	δ_{61}
δ_{30}	δ_{51}	δ_{80}	δ_{60}
δ_{30}	δ_{52}	δ_{80}	δ_{60}
δ_{31}	-	-	δ_{60}

Table 4.2: Table of values of the Function $f_3(x_5, \phi_8)$

A set of such rules *specifies* a deterministic finite automaton that is better represented with a *transition table* (cf. [?] for a presentation of the relation between the TOT and the Automata Theory). Within the TOM4 modeling framework, such an automaton is defined in a TOM4D behavioral model.

The TOM4D methodology *recommends* the building of tables as Table 4.2 *before* modeling the behavior of a dynamic process with the TOM4D *behavioral model formalism* in order to explicits some logical constraints about this behavior. As a consequence, functions containing at least one integrator will be represented in the functional model *and* the behavioral model: this means simply that such a function will be implemented with an automata that is specified in a behavioral model.

In a symmetrical way, an algebraic representation of a function $x_2 = f(x_1)$ defined on $\Delta_1 \times \Delta_2$, $\Delta_1 = \{\delta_{1i}\}_{i=1 \ldots n_i}$, $\Delta_2 = \{\delta_{2j}\}_{j=1 \ldots n_j}$, can be represented with a TOM4D behavioral model. This corresponds to the classical *space to time* transformation of the Automata Theory. Within the framework of the TOT, this transformation is based on the transformation of a production rule into a sequential binary relation:

- $x_1 = \delta_{1i} \Rightarrow x_2 = \delta_{2j} \longrightarrow r(C_{1i}, C_{2j})$

This transformation is based on a theorem demonstrated in [LG06] that can be intuitively understood with the *Modus Ponens* in mind (cf. equation 4.2) and the interpretation equation 3.8 of a timed observation. As an occurrence of the class $C_i = \{(x_i, \delta_{1i})\}$, a timed observation $C_{1i}(t_1)$ corresponds to the assignation $x_1 = \delta_{1i}$ at time t_1. Similarly, a timed observation $C_{2j}(t_2)$ corresponds to the assignation $x_2 = \delta_{2i}$ at time t_2. As a consequence, if $t_2 > t_1$, the two timed observations $C_{1i}(t_1)$ and $C_{2j}(t_2)$ satisfies the constraints of a binary sequential relation $r(C_{1i}, C_{2j})$. According to the definition 3.5, the relation $r(C_{1i}, C_{2j})$ is then observed at time t_2. By construction, this observed relation corresponds to assignation $x_1(t_1) = \delta_{1i}$ and $x_2(t_2) = \delta_{2j}$: in other words, to each observation of the relation $r(C_{1i}, C_{2j})$ corresponds the application of the

Modus Ponens with a rule $x_1 = \delta_{1i} \Rightarrow x_2 = \delta_{2j}$ with the assignation $x_1 = \delta_{1i}$ made at time t_1 and the assignation $x_2 = \delta_{2j}$ made at time t_2. The temporal aspect of this correspondence increases drastically the complexity of the complete demonstration that would bring nothing here (the interested Reader is invited to refer to [LG06] which is dedicated to this demonstration and its consequences).

For example, let us consider again the function $x_5 = f_1(x_2)$ described by the table 4.3.

$f_1 : \Delta_{x_2} \rightarrow \Delta_{x_5}$		
x_2	$f_1(x_2)$	*Interpretation:* $x_5 = f_1(x_2)$
δ_{20}	δ_{50}	$x_2 = \delta_{20} \Rightarrow x_5 = \delta_{50}$
δ_{21}	δ_{51}	$x_2 = \delta_{21} \Rightarrow x_5 = \delta_{51}$
δ_{22}	δ_{52}	$x_2 = \delta_{22} \Rightarrow x_5 = \delta_{52}$

Table 4.3: Example of the function $x_5 = f_1(x_2)$

Such a function can be represented with the following behavioral model $BM(X_8(t))$ of a dynamic process, $X_8(t) = \{\ x_2(t),\ x_5(t)\}$ for example, which is the structure $< S_8, C_8, R^{s_8} >$ where:

- The set $S_8 = \{\ s_{80} = \{r(C_{20}, C_{50})\},\ s_{81} = \{r(C_{21}, C_{51})\},\ s_{82} = \{r(C_22, C_{52})\}\ \}$ is made of 3 sequential binary relations defining 3 discernible states.

- The set $C_8 = \{$
  $C_{20} = \{(x_2, \delta_{20})\}, C_{21} = \{(x_2, \delta_{21})\}, C_{22} = \{(x_2, \delta_{22})\},$
  $C_{50} = \{(x_5, \delta_{50})\}, C_{51} = \{(x_5, \delta_{51})\}, C_{52} = \{(x_5, \delta_{52})\}$
  $\}$ made of 6 observations classes.

- The empty set $R^{s_8} = \{\phi\}$ containing no predicate because the discernible states of S_8 are not interconnected.

Generally speaking, this *algebraic to automaton transformation* of a TOM4D function model is usually required when modeling and diagnosing a dynamic process. That why the TOT methodology recommends to *always* provide the both representation of a same set of functions.

4.2.6 Dynamic Process Model Formalism

As a consequence, a TOM4D model of a process $X(t)$ is a structure made of a perception model $PM(X(t))$, a structural model $SM(X(t))$, a functional model $FM(X(t))$ and a behavioral model $BM(X(t))$:

Definition 4.5 *Dynamic Process Model Formalism*
A TOM4D model $M(X(t))$ of a dynamic process $X(t)$ is a structure $< PM(X(t)), SM(X(t)), FM(X(t)), BM(X(t)) >$.

In practice, most real-world dynamic processes are usually too complex to allow the building of a behavior model: it is often intractable and when it possible, the resulting behavioral model is generally so huge to be used efficiently. Such cases are current and frequent.

To face this complexity, Experts usually describes complex processes with a set of interconnected components, that is to as a *network* of dynamic processes. The advantage of the description of a complex dynamic process under the form of a network of interconnected components is the *behavior hiding*: each component of a network has its own role in the network (i.e. its goal), it own structure, function and behavior. Each component is a whole entity, and the global behavior of a network results of the way the components are connected together and the local behavior of each component. It is then possible to analyze the global behavior of a network on two points of view:

1. The network point of view. Only the connexion relations between components are considered.

2. The component point of view. Each component is considered as an autonomous system and so, can be studied as a whole entity.

The advantage the behavior hiding is the *locality* property that entails a *monotonic reasoning* in the analysis of the behavior of a network of interconnected components. Such a monotonic reasoning is particularly efficient when analyzing a fault:

- If it has been demonstrated that a component c_i is not faulty, then it can be removed from the set of the possibly faulty components explaining the abnormal behavior of a network.

The *Superposition Theorem* of the TOT allows, given an adequate set $C = \{C_i\}_{i=1 \ldots n}$ of observation classes, to describe any observed process $(X(t), \Theta(X, \Delta))$ as a *network* of observed processes $\bigcup\limits_i (X_i(t), \theta_i(C_i))$.

The important point for a modeling point of vue is that a network results of the simple partitioning of the set Δ in n disjoint sets Δ_i: no condition is imposed on the dynamic process $X(t)$. In other words, the description of a given dynamic process $X(t)$ under the form of a network $\bigcup\limits_i X_i(t)$ of dynamic processes $X_i(t)$ only results of a *modeling* decision.

Such a modeling decision has no impact on the modeling process of the TOM4D methodology but leads to modify the notion of *component* of the structural model in order to introduce some properties of recursion: a component c_i becomes a *model* of a dynamic process $X_i(t)$.

As a consequence, the TOM4D notion of component c_i is to be understand as a *model* $c_i \equiv M(X_i(t))$ according to the TOM4D methodology. So, a set of n components $COMPS = \bigcup\limits_{i=1 \ldots n} COMPS_i(t)$ defines n subsets of the set $X(t)$ of time function so that:

- $X(t) = \bigcup\limits_{i=1 \ldots n} X_i(t) \equiv X = \bigcup\limits_{i=1 \ldots n} X_i$.

In other words, a component c_i is a complete model $M(X_i(t)) = < PM(X_i(t)), SM(X_i(t)), FM(X_i(t)), BM(X_i(t)) >$ or, to end the recursion, a simple abstract entity denoted $c_i \equiv M(\Phi)$.

Similarly, a TOM4D model of a network of dynamic process is the structure resulting of the union of all the models of components: $M(\bigcup\limits_i X_i(t)) = < PM(\bigcup\limits_i X_i(t)), SM(\bigcup\limits_i X_i(t)), FM(\bigcup\limits_i X_i(t)), BM(\bigcup\limits_i X_i(t)) >$.

4.3 Controlling the Semantic

One of the main difficulty with the application of Knowledge Engineering is the analysis of the semantics contained in a knowledge corpus provided by an expert.

By definition, the knowledge engineer is a ”novice” compared with the experts of the domain and consequently, the knowledge engineer has not the necessary distance to analyze the coherence and the scope of a new piece of knowledge. This difficulty increases drastically when working with a dynamic process: it is very easy for a novice to admit propositions that seems physically reasonable but are not coherent with the current version of the knowledge model. Intuitively, it is easy with the formalism of the TOM4D models to introduces erroneous elements some where on a model. Such a error can have strong and heavy consequences on the development of the models development that can lead to expensive efforts in order to correct such errors.

Fortunately, researches in the Knowledge Engineering domain provided tools to facilitate the semantic analysis. The next section briefly describes the tools the TOM4D methodology resorts on, that is to say the CommonKADS template, the Tetrahedron of States and the Predicate Calculus. This latter being well known, it will not be described.

4.3.1 CommonKADS Template

CommonKADS is a Knowledge Base System (KBS) engineering methodology which offers a structured approach in the management of the development of KBSs.

CommonKADS is well known for the fundamental role that this methodology attributes to the ”Conceptual Model of Expertise” (CME) in the development process of a KBS. This model describes the types and structures of the knowledge required to accomplish a particular cognitive 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 implementation-specific term. Thus, this one is an important vehicle for communication with experts and users about the problem-solving aspects of a KBS.

According to CommonKADS, a CME is a three layer model:

1. The lower level is called the *Domain Layer.* It contains all the concepts and the logical relations between them.

2. The middle level is called the *Inference Layer* because it contains inference steps using the domain rules, attributes the semantic roles the domain concept must play in an inference step and organizes inferences and roles in inference structures.

3. The upper layer, the *Task Layer*, defines the methods (i.e. the prototypical algorithms) that can be used to achieved some problem solving goals with the use of the corresponding inference structure.

A CommonKADS Conceptual Model of Expertise (CME) describes the cognitive process an expert uses to solve a given problem with the corresponding domain knowledge. One fundamental property of a CommonKADS CME is that the internal structures of a CME is independent of the expertise domain because it is directly linked with a cognitive task (Diagnosis for example). So CommonKADS provide a set of expertise templates defining a domain schema, an inference structure and the corresponding method for the main cognitive tasks. Such templates being generic (i.e. independent of the domain knowledge), they can be used to accelerate the acquisition and the modeling of the experts’ knowledge given a cognitive task.

Aiming at producing a generic model of an *observed process*, the TOM4D methodology is designed to resort on the CommonKADS CME:

- When the goal of CommonKADS CME is to describe the expert's reasoning when using the domain knowledge about a process to make a cognitive tasks like diagnosis for example, the goal of a TOM4D model is to describe the process model the experts have in mind when doing such a cognitive task.

- The use of a CommonKADS CME will accelerate the construction of a TOM4D model of the corresponding process.

- A TOM4D model can be used to validate the domain knowledge of a CommonKADS Conceptual Model of Expertise or to help its construction.

In particular, the CommonKADS templates are of the main importance to provide an interpretation of the available knowledge about a dynamic process. CommonKADS being centered on the expert's reasoning, the CommonKADS models are necessary but not sufficient to provide a physical interpretation to a set of variables.

For example, considering the didactic example of the hydraulic system, Sachem general template of Figure 4.5) [LG06, LG04] can be used to analysis the available knowledge described in the preceding section (this template is detailed in [LG04]). This template establishes a link between sensors and abstract *Process Phenomena*: the occurrence of a process phenomena can be observed on particular sensors signals. Such an occurrence is associated with a physical transformation of material that is working inside a process. But the concept of Process Phenomena being abstract, only its *effects* can be physically observed. So to observe the occurrence of a process phenomena, the reasoning elaborates *signals* on which particular combination *Signal Events* are looked for. When such a combination matches with a model, an occurrence of *Signal Phenomena* is created. Again, when a specific combination of signal phenomena occurrences are observed on different sensor signals, an occurrence of *Process Phenomena* is created. It is to note that the combination of occurrences of signal events or signal phenomenon can be described with abstract chronicle model of the TOT (cf. [LG04] for examples). This template will be used in the suite of this chapter to analyze the available knowledge about the behavior of the didactic example of the hydraulic process.

Such a template describes the expert's reasoning about physical phenomenon working inside a dynamic process. But it says nothing about the physical *nature* of these phenomena: these elements are provided in the domain layer of a CME. And the CommonKADS provides no tools to help the knowledge engineer to analyse this kind of knowledge.

4.3.2 Tetrahedron of States

This explains why the TOM4D methodology resorts also to the Tetrahedron of States (ToS, figure 4.6) of Rosenberg and Karnopp [RK83] to provide a physical interpretation of the expert's propositions. The ToS describes the fundamental structure of the classical Newtonian physics with a set of four generalized continuous variables:

- e for *effort*,

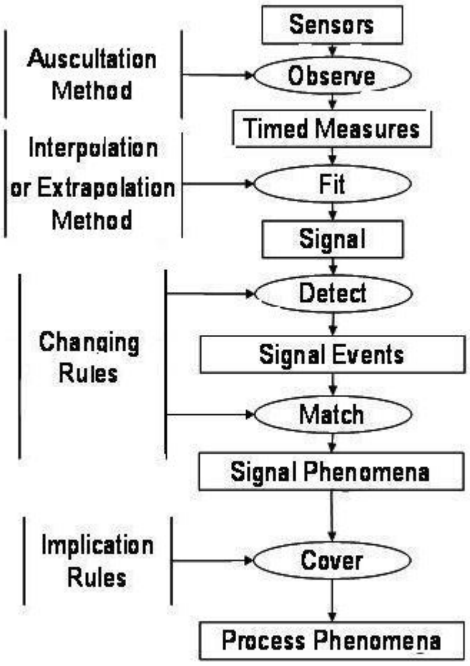


Figure 4.5: Inference structure for diagnosis [LG04]

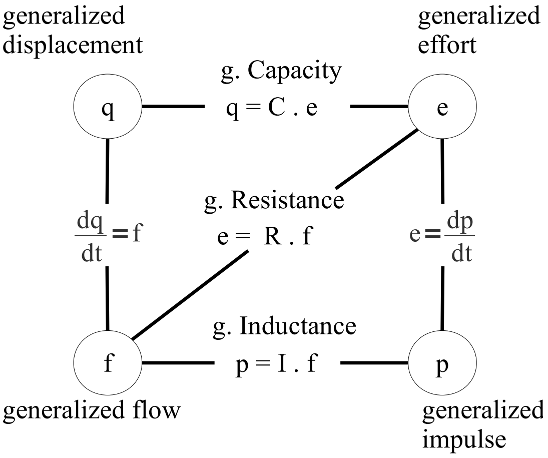


Figure 4.6: Tetrahedron of States (ToS) [RK83]

- f for *flow*,
- p for *impulse*, and
- q for *displacement*.

The ToS defines also three type of constant (figure 4.6):

- C for *Capacity*,
- R for *Resistance*, and
- I for *Inductance*.

These elements are linked together with a set of five binary relations. Three of them are algebraic:

$$q = C.e \tag{4.7}$$

$$e = R.f \tag{4.8}$$

$$p = I.f \tag{4.9}$$

The two others are the usual differential relations:

$$f = \frac{dq}{dt} \tag{4.10}$$

$$e = \frac{dp}{dt} \tag{4.11}$$

This set of binary relations defines three type of generic components: a *Generalized Capacity*, a *Generalized Resistance* and a *Generalized Inductance*.

To use it, the ToS must be instantiated to a physical domains like electromagnetism, fluid dynamics, thermodynamics, etc. For example, in the hydraulic dam domain that concerns this works (cf. figure 4.7, the "Hydraulic ToS" maps a water *flow* to the generalized flow f, the water *volume* to the generalized displacement q, the water *pressure* to the generalized effort "e" and the hydraulic momentum to the generalized impulse p.

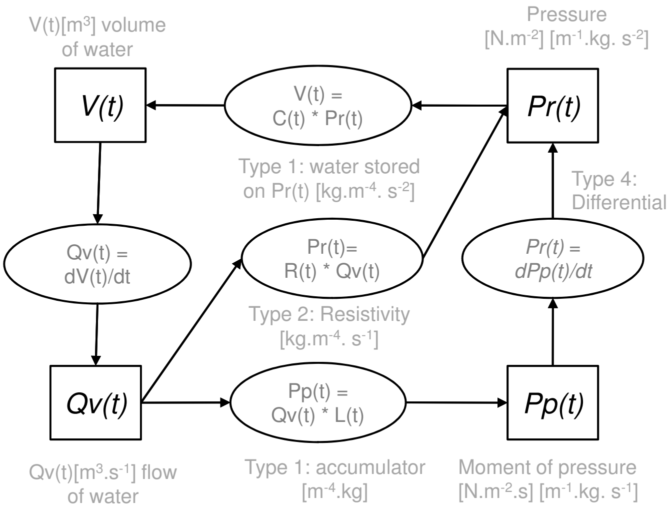

V(t)[m3] volume of water
Pressure [N.m-2] [m-1.kg. s-2]
V(t)
V(t) = C(t) * Pr(t)
Pr(t)
Type 1: water stored on Pr(t) [kg.m-4. s-2]
Type 4: Differential
Qv(t) = dV(t)/dt
Pr(t)= R(t) * Qv(t)
Pr(t) = dPp(t)/dt
Type 2: Resistivity [kg.m-4. s-1]
Qv(t)
Pp(t) = Qv(t) * L(t)
Pp(t)
Qv(t)[m3.s-1] flow of water
Type 1: accumulator [m-4.kg]
Moment of pressure [N.m-2.s] [m-1.kg. s-1]

Figure 4.7: Instantiation of the ToS in the Hydraulic Domain

According to the Hydraulic ToS, any time function $x_i(t)$ of a process $X(t)$, and then any variable x_i of the corresponding set X, will receive a physical dimension and any proposition about the relations between two hydraulic variables can be analyzed according to the Hydraulic ToS. As a consequence, a proposition that don't satisfy one of the five relations of the Hydraulic ToS must be rejected by the knowledge engineer. An instantiated ToS constitutes then a powerful tool to interpret and provide a meaning to the expert knowledge, that is to say to control the introduction of the semantics within the TOM4D modeling process.

In complementary with the ToS, the Formal Logic framework, and more precisely the first order predicate calculus in particular, is also used by the TOM4D methodology as a resource to

Domain	$e \to Pr$	$f \to Q$	$q \to V$	$p \to Pp$
Hydraulic	Pressure (Pa)	flow $(m^3.s^{-1})$	volume (m^3)	$N.m^{-2}.s$

Table 4.4: Physical Dimension of the Variables of the Hydraulic ToS

provide a way to analyze the logical consequences of a ToS validated proposition: a new proposition can satisfy the semantic constrains of the ToS but can be contradictory with the current version of the model. In this latter case, the knowledge engineer must solve the contradiction either with the rejection of the new proposition or with an adequate update of the current version of the model that contains the new proposition. It is to note that using the Formal Logic framework allows the utilization of Reiter's Theory of Diagnosis [Rei87] to facilitate the analysis of the model coherence.

Finally, the combination of the CommonKADS templates, the ToS and the first order predicate calculus constitute a powerful interpretation framework to control the introduction of semantic elements during the modeling process of the TOM4D methodology.

4.4 TOM4D Modeling Process

The aim of the TOM4D modeling process is to produce a generic model of an observed process $(X(t), \Theta(X, \Delta)$ from the available knowledge and data.

The available knowledge contains, by definition, a description of the modeling goal: the assessment of failure linked with the dynamic process exploitation. The TOM4D modeling process is made with three main phases:

- Knowledge Interpretation,
- Process Definition, and
- Generic Modeling.

The figure 4.8 present the global TOM4D modeling process under the form of a CommonKADS-like inference structure (i.e. a precedence graph): ovals represents inference steps and rectangles defines the concepts' roles. In other words, the figure shows the logical dependencies of these three phases. Therefore, the exploitation method of such an inference structure must be defined according to the modeling problem: how the control flow of the modeling process is carried out is not part of this structure. This explains why TOM4D is not a *method* but a *methodology*. For example, a spiral model is particularly appropriate because it is a risk-driven process model that allows an incremental construction of the model. Such a development model being cyclical, each modeling step can require to return back to previous step with the objective of revising the expert's knowledge or modeling decisions.

4.4.1 Knowledge Interpretation

The objective of the *Knowledge Interpretation* step is to define a scenario model $M(\Omega)$ given the available knowledge, a set Ω of sequences ω_i of variable values over time and a CommonKADS template of the cognitive task (the Sachem template of figure 4.5 for example). The TOM4D methodology being firstly concerned with dynamic process, the temporal behavior of the process

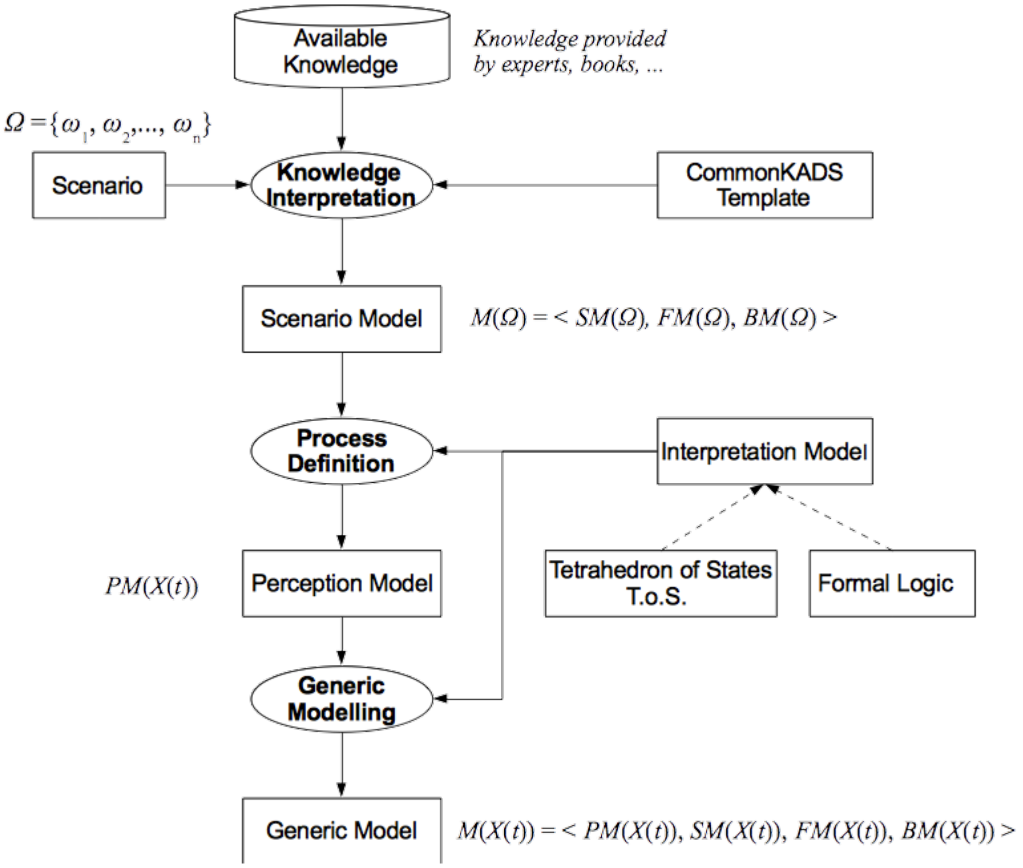


Figure 4.8: General structure of the TOM4D modeling process.

is of the main importance in the analysis. Because this behavior is, generally speaking, poorly described by the available knowledge, the scenario Ω is a fundamental input to the modeling process. Each sequence ω_i of Ω is an ordered set of measure of the values some process variables takes over time. These suites of timed measures describes a particular evolution of the process.

As a consequence, Ω constitutes a partial description of the observed process and so, defines a primary set of functions, variables, constants and stochastic clocks. The identification of these elements is made through the available knowledge that is interpreted with a CommonKADS template. This template describes the usual cognitive task the experts do when analyzing the behavior of the dynamic process. The analysis of the relations between the primary elements leads to a set of primary knowledge representations that are organized in a primary model called the *scenario model* $M(\Omega) = < SM(\Omega), FM(\Omega), BM(\Omega) >$ linking together a structural model $SM(\Omega)$, a functional model $FM(\Omega)$ and a behavioral one $BM(\Omega)$ of the process according to the scenario Ω. Again, by construction,$M(\Omega)$ can not be complete: it is restricted to what appears in the sequences ω_i of Ω. But, if Ω is representative enough of the behavior of the observed process, such a model is sufficient to provide a first *vision* of the process to be modeled. Generally speaking, the representativeness of Ω is easy to assess. Ω being an extraction of a database (or a data log), it is necessary and sufficient to verify that the extracted sequences contain the most typical evolutions of the process $X(t)$ that the program $\Theta(X, \Delta)$ aims at describing with timed observations.

4.4.2 Process Definition

The second main step of the TOM4D modeling process is called the *Process Definition* since it aims at providing the boundary of the process $X(t)$ in terms of a set of time functions $x_i(t)$, the goals of the process operations and its normal and abnormal operating modes.

The actual process is then restricted to a particular set $X(t) = \{x_i(t)\}$ of time function $x_i(t)$ and its operational goals are provided with a set of conjunctions of propositions of the following forms where the symbol $\neg$ denote the logical "not":

- Positive Goal: $\forall t, x_i(t) \geq \Phi_i$
- Negative Goal: $\forall t, \neg x_i(t) \geq \Phi_i$

Clearly, a negative goal is typically a direct formulation of an abnormal behavior linked with the exploitation of the process $P(t)$. The operating modes are also represented with a set of conjunctions of the same form. The set of conjunctions is partitioned in two sets: the set of the desired and the set of the undesired modes, respectively called the normal behavior set and the abnormal behavior set according to Reiter's theory of Diagnosis [Rei87].

The input knowledge of the Process Definition step are the scenario model $M(\Omega)$ resulting of the Knowledge Interpretation step and the conceptual frameworks of the Formal Logic and the Tetrahedron of States (ToS) [RK83]. These two frameworks constitutes the only semantic contexts allowing the logical and the physical interpretation of the modeling symbols used to denote the variables and the constants defined in the scenario model $M(\Omega)$. The role of these frameworks is to provide the set of laws allowing the knowledge engineer to control the representation of the semantic in the model and consequently the interpretation of the binary relations of the model. This step being an important and delicate point in the TOM4D methodology, it will be discussed with more details when the TOM4D modeling process will be applied on the didactic example.

The output of the Process Definition step is the *Perception Model* $PM(X(t))$ of the dynamic process since it defines the way the process is perceived by the experts: nothing but what can be derived from this model can be taken into account the structural, the functional and the behavioral model of the process.

In other words, $PM(X(t))$ defines the level of abstraction the expert use to reason about the process $X(t)$.

4.4.3 Generic Modeling

The last step, the *Generic Modeling*, defines the set X=$\{x_i\}$ of the variables with their definition domain Δ_{x_i}=$\{delta_i^{x_i}\}$, identifies the corresponding sets of components, observation classes and logical relations between the constants of the definition domain of the variable, and distributes the representation of the pertinent binary relations over the three models, that is to say the structural model $SM(Pr(t))$, the functional model $FM(Pr(t))$ and the behavioral model $BM(Pr(t))$.

The objective is then to define a model $M(X(t)) = < PM(X(t))$, $SM(X(t))$, $FM(X(t))$, $BM(X(t)) >$ a type of process that is coherent with the scenario model $M(\Omega)$, but that generalizes it: this is the meaning of the usage of the *generic* attribute to qualify the generality and the abstraction levels of the resulting model $M(X(t))$.

The Generic Modeling step is accomplished using the Perception Model $PM(X(t))$ and the available knowledge, according to the representation and the interpretation laws of the Formal Logic and the ToS frameworks. These frameworks allow the systematic exploration of the whole semantic and syntactic spaces that constitutes the global modeling space:

- the semantic space is defined with the physical dimension of the variables x_i (typically according to the International System of Units), and
- the syntactic space is defined as the matrix of all the pairs (a, b) that can be made with the alphabet of the symbols used to represents the knowledge.

The next section presents the application of the TOM4D methodology on the didactic example presented in the precedent chapter, the hydraulic system of [CPR00]. This example aims to illustrate the different points discussed during the presentation of the TOM4D methodology.

4.5 Application on a Didactic Example

The TOM4D methodology is illustrated with the didactic example of a hydraulic system described in [CPR00].

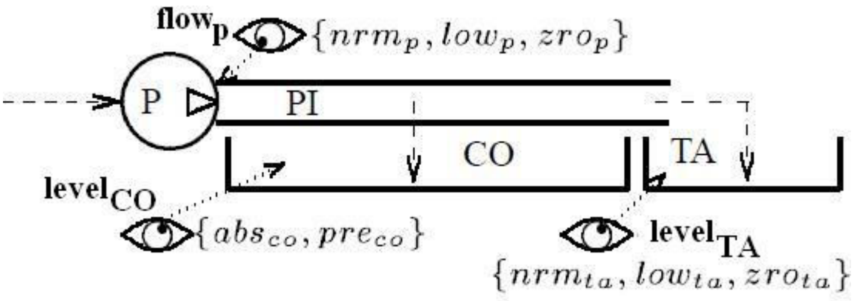


Figure 4.9: The hydraulic system of [CPR00]

Figure 4.9 provides a schematic representation of the hydraulic system. Let us consider that the available knowledge is the following text extracted from [CPR00], only the text format has been changed:

1. The system is formed by a *pump* P which delivers *water* to a *tank* TA via a *pipe* PI; another tank CO is used as a *collector* for water that may leak from the pipe.
2. For the sake of simplicity we assume that the pump is always on and supplied of water.
3. The pump P has three modes of behavior:
   - *ok* (the pump produces a normal output flow),
   - *leaking* (it produces a low output flow) and
   - *blocked* (no output flow).
4. The pipe PI can be:

- *ok* (delivering to the tank the water it receives from the pump) or
- *leaking* (in this case we assume that it delivers to the tank a low output when receiving a normal or low input, and no output when receiving no input).

5. The tanks TA and CO are always in mode ok, i.e., they simply receive water.

6. We assume that three sensors are available (see the eyes in Figure 4.9):

   - $flow_p$ measures the flow from the pump, which can be normal (nrm_p), low (low_p), or zero (zro_p);
   - $level_{TA}$ measures the level of the water in TA, which can be normal (nrm_{ta}), low (low_{ta}), or zero (zro_{ta});
   - $level_{co}$ records the presence of water in CO, either present (pre_{co}) or absent (abs_{co})

Let us supposes that this textual description of the hydraulic system and the figure 4.9 constitutes the complete available knowledge about the hydraulic system to model. Let us supposes also that these elements are given with the aim of implementing a diagnosis taks. To this aim, let us suppose that this description is provided with the purpose of a diagnosis cognitive task. The next chapter proposes the algorithms designed to use the TOM4D models that will be made all over the the next sections of this chapter.

According to the modeling process of figure 4.8, the application of the TOM4D methodology requires the definition of a method that organizes the three main steps that are:

1. Knowledge Interpretation. This step aims at defining a first model that corresponds to the available knowledge. This step defines the structure of the process to model and the scientific domain that is concerned under the form of a *scenario model* $M(\Omega)$.

2. Process Definition. This important step defines the way the process is perceived by the experts under the form of a set of time functions $X(t)$ and the operating goals of the process. Given the scenario model $M(\Omega)$, this step produces the *Perception Model* that defines the variable set X and a formal expression of the operating goals. This step uses the interpretation models that are the adequate ToS to introduce a physical interpretation of the variables and the goals, and the Predicate Calculus as a formalism to analyze the consistency of the experts propositions.

3. Generic Modeling. This last step aims at modeling the process under the form of the structure $M(X(t)) = < PM(X(t)), SM(X(t)), FM(X(t)), BM(X(t)) >$ that has been define in the section 4.2.

This didactic example is simple enough to apply this process sequentially, so nothing special has to be defined.

4.5.1 Step 1: Knowledge Interpretation

The interpretation of the figure 4.9 and the 6 points of the text is made according to the CommonKADS template of figure 4.5.

This template establishes a logical link between *Sensors* and *Process Phenomena.* The point 6 of the text defines three sensors providing *Timed Measures* under the form of abstract values. These sensors are associated with the four elements that are explicitly defined as *components* in the point 1, the points 2, 3, 4 and 5 defining their *mode of behavior*:

- Component c_1: The name of c_1 is *PI*, its nature is to be a *pipe.* c_1 has two modes of behavior: *ok* and *leaking*. There is no senor associated with it.
- Component c_2: The name of this component is *P*, its nature is to be a *pump* that has three modes of behavior: *ok*, *leaking* and *blocked.* c_2 is associated with the sensor $flow_p$ that measures over time the *water flow* of P and provides three possible values: nrm_p denoting the *normal* value of the water flow, low_p denoting the *low* value and zro_p denoting the *zero* value.
- Component c_3: The name of c_3 is TA, its nature is to be a *tank* that has only one mode denoted *ok*. c_3 is associated with the sensor $level_{TA}$ that measures over time the *water level* contained in c_3. $level_{TA}$ provides three possible values: nrm_{TA} denoting the *normal* value of the water level, low_{TA} denoting the *low* value and zro_{TA} denoting the *zero* value.
- Component c_4: The name of c_4 is CO, its nature is to be a *tank* with only one mode of behavior *ok*. c_4 is associated with the sensor $level_{CO}$ that measures over time the *presence* of water in c_4. $level_{CO}$ provides two possible values: pre_{CO} denoting that water is *present* in c_4 and abs_{CO} denoting the *absence* of water.

The CommonKADS template of figure 4.5 allows to consider that a *sensor* is a type of component. So three more components can be identified:

- Component c_5: The name of c_5 is $flow_p$, its nature is to be a *sensor.* c_5 provides three values nrm_p (normal), low_p (low) and zro_p (zero).
- Component c_6: The name of c_6 is $level_{CO}$, its nature is to be a *sensor* that provides two values: pre_{CO} (presence) and abs_{CO}.
- Component c_7: The name of this component is $level_{TA}$, its nature is to be a *sensor* that provides three values: nrm_{TA} (normal), low_{TA} (low) and zro_{TA} (zero).

The CommonKADS template of figure 4.5 provides no means to interpret the term *mode of behavior.* Nevertheless, it can noticed that only c_1 and c_2, the *pipe* and the *pump*, have at least two possible values of *mode of behavior.* A tank (i.e. c_3 and c_4) has only one value, and nothing is said about the sensors. Recalling that the multi-value variable principle of the TOT imposes at least two values for a variable, it is then simple, at this step of modeling, to associate a variable to the *mode of behavior* of c_1 and c_2, even if this notion is still undefined.

But a modeling choice must be made about the other components that have either one value, *ok* (the tanks), or no value at all (the sensors). It is usual to take the decision that simplifies the model: if such a decision leads to some problems, it is always possible to come back to take another modeling decision. So no variable will be associated with the *mode of behavior* of all the components except c_1 and c_2. Such a modeling decision means that the components never change of *mode of behavior.*

Variable	**Possible Values**	**Nature**
$mode_{PI}$	$ok, leaking$	*modes of behavior* of a Pipe
$mode_P$	$ok, leaking, blocked$	*modes of behavior* of a Pump
$flow_p$	zro_p, low_p, nrm_p	Values of a Sensor
$level_{CO}$	abs_{CO}, pre_{CO}	Values of a Sensor
$level_{TA}$	$zro_{TA}, low_{TA}, nrm_{TA}$	Values of a Sensor

Table 4.5: Definition of the Concrete Variables

As a consequence, five variables can be *a priori* identified (cf. Table 4.5).

An important point of the TOM4D methodology is the *Symbol Driven Modeling principle.* The role of this principle is to avoid the uncontrolled introduction of semantic elements in the model by the use of anonymous variables and values. This application of this principle leads to the construction of the Table 4.6 where abstract denominations are used to define the variables and their possible values.

Comp.	**Variable**	**Definition Domain**	**Respective Value Name**
c_1	x_1	$\Delta_{x_1} = \{\delta_{10}, \delta_{11}\}$	$leaking, ok$
c_2	x_2	$\Delta_{x_2} = \{\delta_{20}, \delta_{21}, \delta_{22}\}$	$blocked, leaking, ok$
c_5	x_5	$\Delta_{x_5} = \{\delta_{50}, \delta_{51}, \delta_{52}\}$	zro_p, low_p, nrm_p
c_6	x_6	$\Delta_{x_6} = \{\delta_{60}, \delta_{61}\}$	pre_{CO}, abs_{CO}
c_7	x_7	$\Delta_{x_7} = \{\delta_{70}, \delta_{71}, \delta_{72}\}$	$zro_{TA}, low_{TA}, nrm_{TA}$

Table 4.6: Definition of The Abstract Variables

According to the *Spatial Discretization Principle*, the definition domains of the sensor's variables x_5, x_6 and x_7 allows to identify five abstract thresholds (cf. Table 4.7. The values of the variables x_1 and x_2 denoting *modes of behavior*, no thresholds are defined.

Variable	**Definition Domain**	**Respective Abstract Ranges**
x_5	$\Delta_{x_5} = \{\delta_{50}, \delta_{51}, \delta_{52}\}$	$]-\infty, \Psi_{51}[$, $[\Psi_{51}, \Psi_{52}[$, $[\Psi_{52}, +\infty[$
x_6	$\Delta_{x_6} = \{\delta_{60}, \delta_{61}\}$	$]-\infty, \Psi_{61}[$, $[\Psi_{61}, +\infty[$
x_7	$\Delta_{x_7} = \{\delta_{70}, \delta_{71}, \delta_{72}\}$	$]-\infty, \Psi_{71}[$, $[\Psi_{71}, \Psi_{72}[$, $[\Psi_{72}, +\infty[$

Table 4.7: Definition of The Abstract Thresholds

The interpretation of the text is sufficient to define a first structural model $SM(\Omega)$, before providing the Ω sequence (cf. figure 4.10).

Formally, according to the definition 4.2 TOM4D methodology, the structural model $SM(\Omega)$ is a structure $< COMPS, R^p, R^x >$ where:

- $COMPS = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$
- $R^p = \{out_1(c_2) = in_1(c_1), out_2(c_2) = in(c_5), out_1(c_1) = in(c_3), out_2(c_1) = in(c_4), out(c_3) = in(c_7), out(c_4) = in(c_6), \}$
- $R^x = \{x_1 = in_2(c_1), x_2 = in(c_2), x_5 = out(c_5), x_6 = out(c_6), x_7 = out(c_7)\}$

A first structural model being made, it is then possible to analyze the relations between the values of the variables. The structural model of figure 4.10 allows the identification of the following functions (cf. figure 4.11):

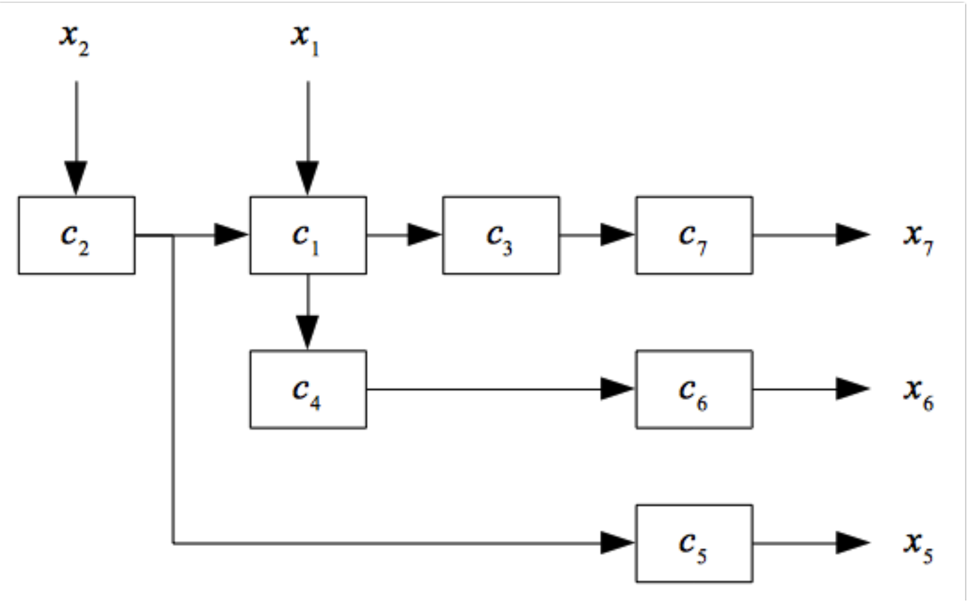


Figure 4.10: Structural Model $SM(\Omega)$

- $x_5 = f1(x2)$: there is a relation between the values of x_5 and those of x_2.
- $x_7 = f2(x_5)$: there is a relation between the values of x_7 and those of x_5.
- $x_6 = f3(x_5, x_1)$: there is a relation between the values of x_6 and those of x_5 and x_1.

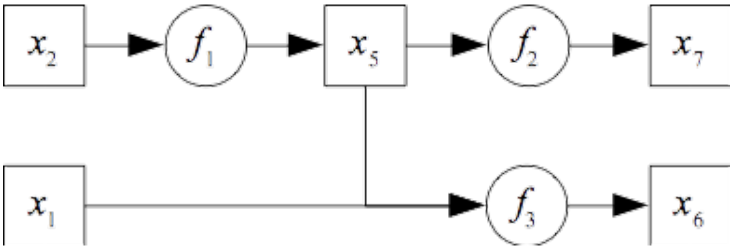


Figure 4.11: Functional Model $FM(\Omega)$

Formally, according to the definition 4.3 of TOM4D methodology, the functional model $FM(\Omega)$ is a structure $< \Delta, F, R^f >$ where:

- $\Delta = \underset{i=1,2,5,6,7}{\cup} \Delta_{x_i}$ (the Δ_{x_i} are given in the table 4.6).
- $F = \{\ f_1 : \Delta_{x_2} \rightarrow \Delta_{x_5},\ f_2 : \Delta_{x_5} \rightarrow \Delta_{x_7},\ f_3 : \Delta_{x_1} \times \Delta_{x_5} \rightarrow \Delta_{x_6}\ \}$.
- $R^f = \{\ x_5 = f_1(x_2),\ x_7 = f_2(x_5),\ x_6 = f_3(x_1, x_5)\ \}$.

Naturally, at this step of the modeling process, the function f_1, f_2 and f_3 cannot be entirely specified. Nevertheless, considered together, the points 3 and 5 of the original text allow to identify the relations between the *modes of behavior* of c_2 (the *pump*, that is to say the values of x_2, and the values of x_5 (i.e. $flow_p$). Here again, the template of 4.5 provides a meaning to the relations between the values: a sensor measuring a phenomena, the values it provide are consequences of the phenomena. This allows to define the followings relations between x_2 and x_5:

- $x_2 = \delta_{22} \Rightarrow x_5 = \delta_{32}$ (c_2 is *ok* → $flow_p$ is nrm_p)
- $x_2 = \delta_{21} \Rightarrow x_5 = \delta_{31}$ (c_2 is *leaking* → $flow_p$ is low_p)
- $x_2 = \delta_{20} \Rightarrow x_5 = \delta_{30}$ (c_2 is *blocked* → $flow_p$ is zro_p)

A similar analysis can be made with the *modes of behavior* of the *pipe* c_1 (i.e. x_1) and the values of x_6 (i.e. $level_{CO}$):

- $x_1 = \delta_{11} \Rightarrow x_6 = \delta_{61}$ (c_1 is *ok* → $level_{CO}$ is abs_{CO})
- $x_1 = \delta_{10} \wedge x_5 = \delta_{51} \Rightarrow x_6 = \delta_{60}$ (c_1 is *leaking* and $flow_p$ is low_p → $level_{CO}$ is pre_{CO})
- $x_1 = \delta_{10} \wedge x_5 = \delta_{52} \Rightarrow x_6 = \delta_{60}$ (c_1 is *leaking* and $flow_p$ is nrm_p → $level_{CO}$ is pre_{CO})

But nothing is said about the relation between the values of x_5 ($flow_p$) and x_6 ($level_{CO}$) and those of x_7 ($level_{TA}$): the usual interpretation of of the terms *flow* and *level* suggests a derivative relation where the time have a place. On the other hand, the structural model (cf. figure 4.10) suggests that the output flow of the *pump*, measured by x_5, is decomposed in two flows: one for c_3 (the *tank TA*) and the other for c_4 (the *tank CO*). So, there could be a more complex relation than the functional model describes at this step of the modeling process.

To go further, a *scenario* must be constructed to describes a part of the behavior of the global process. Here again, the template of 4.5 helps to this aim: a *sensor* being an instrument that provides *measures* over time about a *process phenomena*, these measures can be represented with a time function that constitute a *signal* when ordering the timed values according to they time-stamp. So, a naive but natural interpretation of the 6 points of the text allows to build the following first scenario ω_1 called the *Licking Pipe* scenario:

1. Initially, at $t = t_0$, the pump is blocked, $x_2(t_0) = \delta_{20}$, the pipe PI is ok ($x_1(t_0) = \delta_{11}$), the tanks TA and CO are empty: $x_7(t_0) = \delta_{70}$ and $x_6(t_0) = \delta_{61}$. In such a situation, $x_5(t_0) = \delta_{50}$ ($flow_p$ is zro_p).
2. At $t = t_1$, the pump is turn on: $x_2(t_1) = \delta_{22}$.
3. At $t = t_2$, the output flow of the pump is low and enters in the pipe ($x_5(t_2) = \delta_{51}$). The level in the tank TA increases.
4. At $t = t_3$, the output flow of the pump reaches the its normal value: $x_5(t_3) = \delta_{52}$.
5. At $t = t_4$, the level in the tank TA achieves the low level : $x_7(t_4) = \delta_{71}$.
6. At $t = t_5$, the level in the tank TA achieves the normal level: $x_7(t_5) = \delta_{72}$.
7. At $t = t_6$, the pipe PI is leaking ($x_1(t_6) = \delta_{10}$): a part of the output water flow of the pump goes into the tank CO (i.e. its level increases).
8. At $t = t_7$, some water can be seen in the tank CO: $x_6(t_7) = \delta_{60}$.

This scenario supposes that the dynamic of the *pump* is the lower of the global process.

A second naive scenario ω_2 can be made, called the *blocked Pump* scenario:

1. Initially, at $t = t_9$, the pump is blocked, $x_2(t_9) = \delta_{20}$, the pipe PI is ok ($x_1(t_9) = \delta_{11}$), the tanks TA and CO are empty: $x_7(t_9) = \delta_{70}$ and $x_6(t_9) = \delta_{61}$. In such a situation, $x_5(t_9) = \delta_{50}$ ($flow_p$ is zro_p).
2. At $t = t_{10}$, the pump is turn on: $x_2(t_{10}) = \delta_{22}$.
3. At $t = t_{11}$, the output flow of the pump is low and enters in the pipe ($x_5(t_{11}) = \delta_{51}$). The level in the tank TA increases.
4. At $t = t_{12}$, the output flow of the pump reaches the its normal value: $x_5(t_{12}) = \delta_{52}$.
5. At $t = t_{13}$, the level in the tank TA achieves the low level : $x_7(t_{13}) = \delta_{71}$.
6. At $t = t_{14}$, the level in the tank TA achieves the normal level: $x_7(t_{14}) = \delta_{72}$.
7. At $t = t_{15}$, the pump P is leaking ($x_2(t_{15}) = \delta_{21}$): the pump output water flow decreases.
8. At $t = t_{16}$, the output flow of the pump enters again in the low range: $x_5(t_{16}) = \delta_{51}$.
9. At $t = t_{17}$, the pump P is blocked ($x_2(t_{17}) = \delta_{20}$): the pump output water flow continues to decrease.
10. At $t = t_{18}$, the flow in the pipe enters again in the zero range: $x_5(t_{18}) = \delta_{50}$.

These scenario made some hypothesis about the dynamic of the hydraulic process. But they allow to identify an important point: only the variables associated with sensors can be used to defined observations classes, that is to say the variables x_5, x_6 and x_7.

Nevertheless, with the aim of illustrating the behavioral model formalism, let us suppose that all the classes are observable. To build a set C of observation classes, the simplest consists in defining an observation class as a singleton for each pair (x_i, δ_{ij}), (cf. Table 4.8). In this table, the classes are denoted $C_{ij} = \{(x_i, \delta_{ij})\}$ where i denotes the variable index and ij the index of the corresponding value δ_{ij}.

Variable	**Definition Domain**	**Observation Classes**
x_1	$\Delta_{x_1} = \{\delta_{10}, \delta_{11}\}$	$C_{10} = \{(x_1, \delta_{10})\}$, $C_{11} = \{(x_1, \delta_{11})\}$
x_2	$\Delta_{x_2} = \{\delta_{20}, \delta_{21}, \delta_{22}\}$	$C_{20} = \{(x_2, \delta_{20})\}$, $C_{21} = \{(x_2, \delta_{21})\}$, $C_{22} = \{(x_2, \delta_{22})\}$
x_5	$\Delta_{x_5} = \{\delta_{50}, \delta_{51}, \delta_{52}\}$	$C_{50} = \{(x_5, \delta_{50})\}$, $C_{51} = \{(x_5, \delta_{51})\}$, $C_{52} = \{(x_5, \delta_{52})\}$
x_6	$\Delta_{x_6} = \{\delta_{60}, \delta_{61}\}$	$C_{60} = \{(x_6, \delta_{60})\}$, $C_{61} = \{(x_6, \delta_{61})\}$
x_7	$\Delta_{x_7} = \{\delta_{70}, \delta_{71}, \delta_{72}\}$	$C_{70} = \{(x_7, \delta_{70})\}$, $C_{71} = \{(x_7, \delta_{71})\}$, $C_{72} = \{(x_7, \delta_{72})\}$

Table 4.8: Definition of The Observation Classes

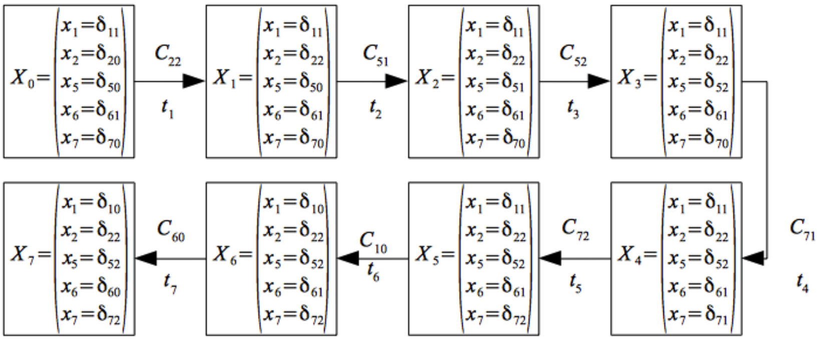
Given the table 4.8, the two scenarii can be written in therm of occurrences of observation classes:

1. $\omega_1 = \{C_{22}(t_1), C_{51}(t_2), C_{52}(t_3), C_{71}(t_4), C_{72}(t_5), C_{10}(t_6), C_{60}(t_7)\}$.
2. $\omega_2 = \{C_{22}(t_{10}), C_{51}(t_{11}), C_{52}(t_{12}), C_{71}(t_{13}), C_{72}(t_{14}), C_{21}(t_{15}), C_{51}(t_{16}), C_{20}(t_{17}), C_{50}(t_{18})\}$.

It is recalled also that, according to the *Spatial Discretization Principle* of the TOT, the interpretation of an occurrence $C_{ij}(t_k)$ is $x_i(t_k) = \delta_{ij}$ (i.e. $C_{ij}(t_k) \equiv x_i(t_k) = \delta_{ij}$), meaning that the corresponding time function $x_i(t)$ enters in a particular range that can be denoted δ_{ij}. As a consequence, the first observations of the two scenarii, that is to say $C_{70}(t_0)$, $C_{61}(t_0)$, $C_{50}(t_0)$ for ω_1 and $C_{70}(t_9)$, $C_{61}(t_9)$, $C_{50}(t_9)$, cannot be written by a safe program $\Theta(X, \Delta)$ because the time functions $x_5(t)$, $x_6(t)$ and $x_7(t)$ does not cross any threshold until Ψ_{51} for x_5, Ψ_{71} for x_7 and Ψ_{60} for x_6(cf. table 4.7).

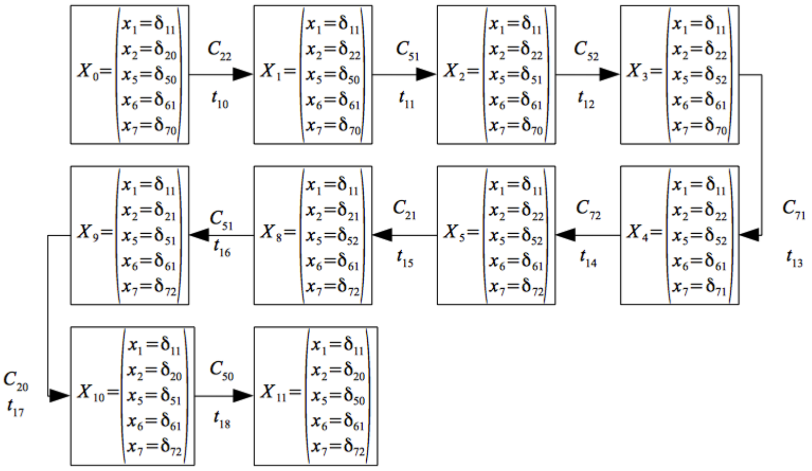
But to illustrate the behavioral modeling, let us define the initial state, denoted s_0, where $x_1 = \delta_{11}$, $x_2 = \delta_{20}$, $x_5 = \delta_{50}$, $x_6 = \delta_{60}$ and $x_7 = \delta_{70}$. Let us then define also a vector, denoted X and defined on $\Delta_{x_1} \times \Delta_{x_2} \times \Delta_{x_5} \times \Delta_{x_6} \times \Delta_{x_7}$ of dimension 5 corresponding to the variables x_1, x_2, x_5, x_6 and x_7 so that each discernible state s_k is associated with one an only one value X_k and reversely. Let us finally denote X_k, $k \in N$, the values of this vector so that a partcular value X_k corresponds to: $x_1 = \delta_{1i}$, $x_2 = \delta_{2j}$, $x_5 = \delta_{5l}$, $x_6 = \delta_{6m}$ and $x_7 = \delta_{7n}$ (the initial state is denoted X_0).

The figures 4.12 and 4.13 proposes a graphical representation of the sequences of discernible states entail by the sequences ω_1 and ω_2 of timed observations. With these figures, it is easy to build behavioral model of figure 4.14 of the set $\Omega = \{\omega_1, \omega_2\}$.



$X_0=$ $x_1=\delta_{11}$ $x_2=\delta_{20}$ $x_5=\delta_{50}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{22} t_1
$X_1=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{50}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{51} t_2
$X_2=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{51}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{52} t_3
$X_3=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{71} t_4
$X_4=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{71}$
C_{72} t_5
$X_5=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{10} t_6
$X_6=$ $x_1=\delta_{10}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{60} t_7
$X_7=$ $x_1=\delta_{10}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{60}$ $x_7=\delta_{72}$

Figure 4.12: Graphical Representation of the Behavioral Model $BM(\omega_1)$



$X_0=$ $x_1=\delta_{11}$ $x_2=\delta_{20}$ $x_5=\delta_{50}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{22} t_{10}
$X_1=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{50}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{51} t_{11}
$X_2=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{51}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{52} t_{12}
$X_3=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{71} t_{13}
$X_4=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{71}$
C_{72} t_{14}
$X_5=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{21} t_{15}
$X_8=$ $x_1=\delta_{11}$ $x_2=\delta_{21}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{51} t_{16}
$X_9=$ $x_1=\delta_{11}$ $x_2=\delta_{21}$ $x_5=\delta_{51}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{20} t_{17}
$X_{10}=$ $x_1=\delta_{11}$ $x_2=\delta_{20}$ $x_5=\delta_{51}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{50} t_{18}
$X_{11}=$ $x_1=\delta_{11}$ $x_2=\delta_{20}$ $x_5=\delta_{50}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$

Figure 4.13: Graphical Representation of the Behavioral Model $BM(\omega_2)$

Formally, according to the definition 4.4 of TOM4D methodology, the behavioral model $BM(\Omega)$ of the Ω sequence is the structure $< S, C, R^s >$ where:

- $S = \{$
  $s_0 = \{r(C_\phi, C_{22})\}$, $s_1 = \{r(C_{22}, C_{51})\}$, $s_2 = \{r(C_{51}, C_{52})\}$, $s_3 = \{r(C_{52}, C_{71})\}$,

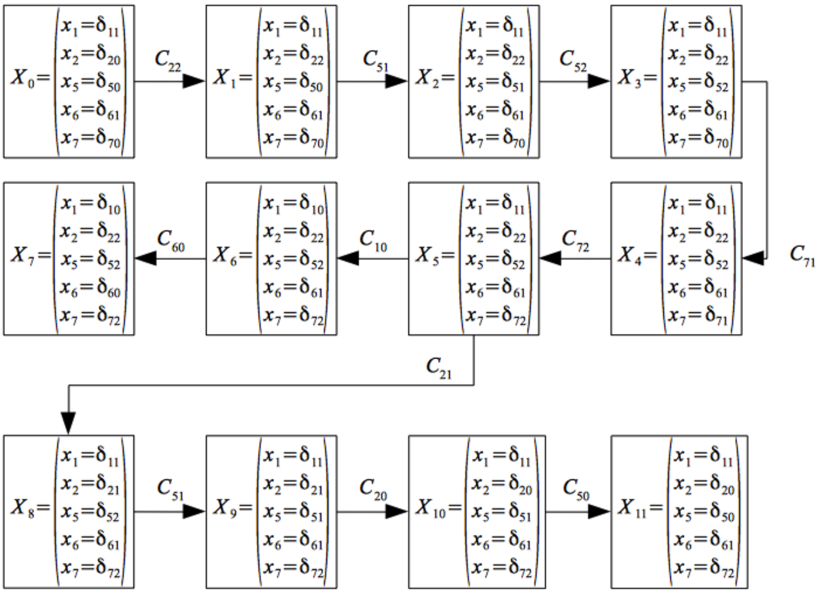

$X_0=$ $x_1=\delta_{11}$ $x_2=\delta_{20}$ $x_5=\delta_{50}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{22}
$X_1=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{50}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{51}
$X_2=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{51}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{52}
$X_3=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{70}$
C_{71}
$X_4=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{71}$
C_{72}
$X_5=$ $x_1=\delta_{11}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{10}
$X_6=$ $x_1=\delta_{10}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{60}
$X_7=$ $x_1=\delta_{10}$ $x_2=\delta_{22}$ $x_5=\delta_{52}$ $x_6=\delta_{60}$ $x_7=\delta_{72}$
C_{21}
$X_8=$ $x_1=\delta_{11}$ $x_2=\delta_{21}$ $x_5=\delta_{52}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{51}
$X_9=$ $x_1=\delta_{11}$ $x_2=\delta_{21}$ $x_5=\delta_{51}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{20}
$X_{10}=$ $x_1=\delta_{11}$ $x_2=\delta_{20}$ $x_5=\delta_{51}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$
C_{50}
$X_{11}=$ $x_1=\delta_{11}$ $x_2=\delta_{20}$ $x_5=\delta_{50}$ $x_6=\delta_{61}$ $x_7=\delta_{72}$

Figure 4.14: Graphical Representation of the Behavioral Model $BM(\Omega)$

$s_4 = \{r(C_{71}, C_{72})\}$, $s_5 = \{r(C_{72}, C_{10})\}$, $s_6 = \{r(C_{10}, C_{60})\}$, $s_7 = \{r(C_{72}, C_{\phi})\}$,
$s_8 = \{r(C_{21}, C_{51})\}$, $s_9 = \{r(C_{51}, C_{20})\}$, $s_{10} = \{r(C_{20}, C_{50})\}$, $s_{11} = \{r(C_{50}, C_{\phi})\}$
$\}$

- $C = \{$
  $C_{10} = \{(x_1, \delta_{10})\}$, $C_{11} = \{(x_1, \delta_{11})\}$,
  $C_{20} = \{(x_2, \delta_{20})\}$, $C_{21} = \{(x_2, \delta_{21})\}$, $C_{22} = \{(x_2, \delta_{22})\}$,
  $C_{50} = \{(x_5, \delta_{50})\}$, $C_{51} = \{(x_5, \delta_{51})\}$, $C_{52} = \{(x_5, \delta_{52})\}$,
  $C_{60} = \{(x_6, \delta_{60})\}$, $C_{61} = \{(x_6, \delta_{61})\}$,
  $C_{70} = \{(x_7, \delta_{70})\}$, $C_{71} = \{(x_7, \delta_{71})\}$, $C_{72} = \{(x_7, \delta_{72})\}$
  $\}$

- $R^s = \{$
  $s_1 = \{(s_0, C_{22})\}$, $s_2 = \{(s_1, C_{51})\}$, $s_3 = \{(s_2, C_{52})\}$, $s_4 = \{(s_3, C_{71})\}$, $s_5 = \{(s_4, C_{72})\}$,
  $s_6 = \{(s_5, C_{10})\}$, $s_7 = \{(s_6, C_{60})\}$,
  $s_8 = \{(s_5, C_{21})\}$, $s_9 = \{(s_8, C_{51})\}$, $s_{10} = \{(s_9, C_{20})\}$, $s_{11} = \{(s_{10}, C_{50})\}$,
  $\}$

Clearly, this model is only a very small part of the complete behavioral model of the hydraulic process: potentially, the *discernible state space* contains $2 \cdot 3 \cdot 3 \cdot 2 \cdot 3 = 108$ states. In other words, the building of the complete behavioral model leads to analyze $108 \cdot 107 = 11\,556$ relations between two different discernible states. And on the other hand, the relations between the variables x_7 ($level_{TA}$), x_5 ($flow_p$) and x_6 ($level_{CO}$) remains unclear. A deeper analysis is then clearly required: this is the role of the two next steps of the TOM4D modeling process.

Nevertheless, even if the behavioral model $BM(\Omega)$ is largely uncompleted, it can be considered as *significant* according two points of view:

- the effect of a *leaking pipe* (x_1) on the *presence* of water in the tank CO (x_6), and
- the effect of a *blocked pump* (x_2) on the *water flow* in the pipe (x_5).

More, it provides dynamical informations about the way some of the timed observations may be scheduled by the process. These elements ar of the most importance to defines the way the expert perceives the process.

4.5.2 Step 2: Process definition

Recalling that the Perception Model $PM(X(t))$ of a dynamic process $X(t)$ is a structure $< X, \Psi, R^q >$ (cf. definition 4.1) where X is a set of variable names, Ψ is a set of thresholds and R^q is a set of logical conjunction representing goals and operating modes, the first step of the Process Definition step consists in the definition of the hydraulic process under the form of a set $X(t)$ of time functions.

The models made in the precedent step allows to immediately identify $x_5(t)$, $x_6(t)$ and $x_7(t)$ as time functions:

- x_5 denotes the water output flow of the pump ($flow_p$): it corresponds then to the Q variable of the hydraulic ToS. The physical dimension of x_5 is then $m^3.s^{-1}$.
- x_6 and x_7 denote the water level of the tanks TA and CO respectively: they correspond then to the V variable of the hydraulic ToS. The physical dimension of x_6 and x_7 is then m^3.

The only equation of the hydraulic ToS (cf. figure 4.7) that is required to define the perception model is then the equation 4.12.

$$Q(t) = \frac{dV(t)}{dt} \tag{4.12}$$

This equation allows to analyse the type of components:

1. c_3 and c_4: a *tank* is a *container*. It is characterized by a capacity, a maximum quantity of water that be contained in it, one input flow and one output flow. The original text says nothing about the capacity and the output flow of the *tanks* TA and CO: this means that they are supposed to be managed so that their *levels* are only the result of their input flows (i.e. the level is proportional with the volume).

2. c_2: a *pump* is a *generator* of a water flow. The point 2 of the original text means then that the *pump* P is always fed with water. In other words, the different values of $flow_p$ (i.e. x_5) does not comes from a lack of water: the terms *bloqued* (δ_{20}) and *leaking* (δ_{21}) concerns only the property of the generator to deliver the always available water. The values of x_5 can then be considered as *equivalent* to the values of x_2.

3. c_1: a *pipe* is a *conduit* or a *tube* that is characterized by the fact that its input flow equals its output flow (i.e. there is no storage). For simplicity reason, let us consider that the *pipe* PI is stiff (no deformation). The term *leaking* (δ_{10}) associated to x_1 means then that c_1 can have two outputs: a normal output flow, corresponding to $out_1(c_1) = in(c_3)$ in $SM(\Omega)$, and a *leaking flow*, corresponding to $out_2(c_1) = in(c_4)$. These two flows are not measured but the values of x_1 are clearly linked with the *leaking flow* of the *pipe*.

This simple analysis allows to propose the goals and the operating modes of the hydraulic process, the set Ψ of thresholds being defined in the table 4.7:

- Operating goal: To maintain the level of the tank TA at a normal level and to maintain empty the tank CO
  - $\forall t \geq t_0, x_7(t) \geq \Psi_{72} \wedge x_6(t) \leq \Psi_{61}$
- Normal operating mode: The output flow x_5 of pump P is normal δ_{52}, the level x_7 in the tank is normal δ_{72} and there is no water in the tank CO (i.e. $x_6 = \delta_{60}$)
  - $\forall t \geq t_0, x_5(t) \geq \Psi_{52} \wedge x_7(t) \geq \Psi_{72} \wedge x_6(t) \leq \Psi_{61}$
- Abnormal operating modes: their are simply the negation of the normal operating mode conjonction.
  - $\forall t \geq t_0, x_5(t) < \Psi_{52} \vee x_7(t) < \Psi_{72} \vee x_6(t) > \Psi_{61}$

To define the hydraulic process $X(t)$ as a set of time functions $x_i(t)$, the status of the variables x_1 and x_2 must be clarified. Their values are defined as *mode of behaviors* but this term is not clearly defined and the hydraulic ToS provides no dimension to the variables x_1 and x_2.

Nevertheless, x_1 can have two values: δ_{10} corresponding to *leaking*, and δ_{11} corresponding to *ok*. These two values are directly linked with those of x_6:

- If the *pipe* is *leaking*, and if $flow_p$ is not null, then some water will fall in the tank CO. In that case, $x_6 = \delta_{60}$ (i.e. pre_{CO}) meaning that there is some water in the tank CO.
- If the pipe *pipe* is *ok*, whatever is $flow_p$, no water will fall in the tank CO. If this tank is empty, then $x_6 = \delta_{61}$ corresponding to abs_{CO}.

Using the formal logic as an interpretation framework, it is then possible to consider pre_{CO} is the opposite of abs_{CO}: $\delta_{60} \equiv \neg\delta_{61}$. As a consequence, x_1 can be considered as a boolean variable taking a value in a boolean set $\Delta_{x_1} = \{\delta_{10}, \delta_{11}\}$ so that: $\delta_{10} \equiv \neg\delta_{11}$. It is then possible to define $x_1(t)$ as a boolean time function (i.e. a double Heaviside step function).

A similar reasoning can be done for $x_2(t)$, except that x_2 can have three values: $\Delta_{x_2} = \{\delta_{20}, \delta_{21}, \delta_{22}\}$ corresponding respectively to *blocked*, *leaking* and *ok*. $x_2(t)$ is then a piecewise constant function.

As a consequence, $x_1(t)$ and $x_2(t)$ can enters in the Perception Model of the hydraulic process (cf. definition 4.6). To provide $PM(X(t))$, let us define the starting time t_0 as the time at which the operations begins. This time is suppose to exist: there is always a time that can be taken as the starting time. Let us also define the time t_i as the time at which the goal is reach for the first time, $t_i \geq t_0$. The delay $t_i - t_0$ corresponds then to the amount of time that is necessary to put in the process in its normal operating mode.

Definition 4.6 $PM(X(t))$
The Perception Model $PM(X(t))$ *of the hydraulic process* $X(t) = \{x_1(t),\ x_2(t),\ x_5(t),\ x_6(t),\ x_7(t)\}$ *is the structure* $< X, \Psi, R^q >$ *where:*

- $X = \{x_1, x_2, x_5, x_6, x_7\}$,

- $\Psi = \{\Psi_{51}, \Psi_{52}, \Psi_{60}, \Psi_{71}, \Psi_{72}\}$
- $R^q = R^{goal} \cup R^n \cup R^{ab}$.
  - $R^{goal} = \{\exists t_0, \exists t_i, t_i \geq t_0, \forall t \geq t_i, x_7(t) \geq \Psi_{72}\}$
  - $R^n = \{\forall t \geq t_i, x_7(t) \geq \Psi_{72} \wedge \forall t \geq t_0, x_6(t) \leq \Psi_{60}\}$
  - $R^{ab} = \{\forall t \geq t_i, x_7(t) < \Psi_{72} \vee \forall t \geq t_0, x_6(t) > \Psi_{60}\}$

R^{goal} defines that the goal of the operations is to have $x_7(t) \geq \Psi_{72}$ (i.e. $level_{TA}$ is nrm_{TA}) from the time t_i. This proposition defines the time t_i. R^n simply expresses that the normal operation is satisfying the main goal ($forallt \geq t_i, x_7(t) \geq \Psi_{72}$) without water in the *tank CO* (i.e. $level_{CO}$ is abs_{CO}). As a consequence, R^{ab} is simply the complement of R^n: from the time the goal has been reached, the process must be operate to continue to satisfy the goal (i.e. it not normal that $level_{TA}$ differs from nrm_{TA}) and the *tank CO* must not contain water. Naturally, without any expert about the hydraulic process of the didactic example, it is not possible to validate $PM(X(t))$: the propositions it contains must be considered as hypothesis.

The perception model $PM(X(t))$ being defined, it is then possible to go in the last modeling step, the *Generic Modeling*.

4.5.3 Step 3: Generic Modeling

The point number 2 of this analysis of the type of component is sufficient to provide a formal specification of the function $x_5 = f1(x_2)$, defined on $\Delta_{x_2} \times \Delta_{x_5}$ of the functional model $FM(\Omega)$ (cf. table 4.9).

$f_1 : \Delta_{x_2} \rightarrow \Delta_{x_5}$		
x_2	$f_1(x_2)$	*Interpretation*
δ_{20}	δ_{50}	*the pump P is blocked* $\Rightarrow$ $flow_p$ *is zero*
δ_{21}	δ_{51}	*the pump P is leaking* $\Rightarrow$ $flow_p$ *is low*
δ_{22}	δ_{52}	*the pump P is ok* $\Rightarrow$ $flow_p$ *is normal*

Table 4.9: Definition of the Generic Function f_1

The other two points leads also to a formalization of the relation between the input flow of the tanks TA and CO. When denoting respectively Q_{TA} and Q_{CO} these flows, the following equations can be written :

$$x_5(t) = Q_{TA}(t) + Q_{CO}(t) \tag{4.13}$$

As a consequence:

$$\begin{array}{rcl} Q_{TA}(t) & = & \frac{d}{dt}x_7(t) \\ Q_{CO}(t) & = & \frac{d}{dt}x_6(t) \end{array} \tag{4.14}$$

That is to say:

$$\begin{array}{rcl} x_7(t) & = & \int Q_{TA}(t)dt \\ x_6(t) & = & \int Q_{CO}(t)dt \end{array} \tag{4.15}$$

Now, to clarify the functional model of the hydraulic process, let us define a new abstract variable $\phi_8 \in [0,1] \subset \Re$ such that:

$$\begin{array}{rcl} x_5(t) & = & (1-\phi_8)\cdot x_5(t) + \phi_8 \cdot x_5(t) \\ Q_{TA}(t) & = & (1-\phi_8)\cdot x_5(t) \\ Q_{CO}(t) & = & \phi_8 \cdot x_5(t) \end{array} \quad (4.16)$$

The role of the variable ϕ_8 is to, eventually, distribute the water flow $x_5(t)$ into the tank *CO*:

$$\begin{array}{rcl} \phi_8 = 0 & \Rightarrow & Q_{CO}(t) = 0 \wedge Q_{TA}(t) = x_5(t) \\ \phi_8 > 0 \wedge \phi_8 < 1 & \Rightarrow & Q_{CO}(t) > 0 \wedge Q_{TA}(t) > 0 \\ \phi_8 = 1 & \Rightarrow & Q_{CO}(t) = x_5(t) \wedge Q_{TA}(t) = 0 \end{array} \quad (4.17)$$

This equation provides then a clear *semantic* to the point 4 of the original text, that is to say to the values of the variable x_1:

- The pipe *PI* (c_1) can be *ok* ($x_1 = \delta_{11}$) means: $\phi_8 = 0$.
- The pipe *PI* (c_1) can be *leaking* ($x_1 = \delta_{10}$) means: $\phi_8 > 0$.

There is then a relation between the values of ϕ_8 and those of x_1. The definition domain Δ_{ϕ_8} of ϕ_8 is given in the table 4.10.

Δ_{ϕ_8}	
Constants	*Interpretation*
δ_{80}	$\phi_8 > 0$
δ_{81}	$\phi_8 = 0$

Table 4.10: Definition Domain Δ_{ϕ_8} of ϕ_8

These relation can then be formally represented with a function f_4, defined on $\Delta_{\phi_8} \times \Delta_{x_1}$ (cf. Table 4.11).

$f_4 : \Delta_{\phi_8} \rightarrow \Delta_{x_1}$		
ϕ_8	$f_4(\phi_8)$	*Interpretation*
δ_{80}	δ_{10}	$\phi_8(t) > 0$ *implies the pipe PI is leaking*
δ_{81}	δ_{11}	$\phi_8(t) = 0$ *implies the pipe PI is ok*

Table 4.11: Definition of the Generic Function f_4

The equations 4.15 becomes:

$$\begin{array}{rcl} x_7(t) & = & (1-\phi_8)\cdot \int x_5(t)dt \\ x_6(t) & = & \phi_8 \cdot \int x_5(t)dt \end{array} \quad (4.18)$$

$x_6(t)$ and $x_7(t)$ being *levels* expressed in meter m and the *flows* $x_5(t)$, $Q_{TA}(t)$ and $Q_{CO}(t)$ being expressed in $m^3.s^{-1}$, the variable name ϕ_8 denotes the inverse of a *surface* an is expressed in m^{-2}:

- $\phi_8 = \delta_{81} \equiv \phi_8(t) = 0$ means that the surface is *infinite*: the surface of the *pipe PI* is sufficient to conduct *all* the water flow that traverses it.

- $\phi_8 = \delta_{80} \equiv \phi_8(t) > 0$ means that the surface of *finite*: the surface of the *pipe PI* is not sufficient to conduct all the water flow that comes into it (i.e. there is a hole in the *pipe* or some water overflows the conduct).

The equations 4.18 establish a clear link between the values of the time functions $x_5(t)$ (the water flow in the pipe PI), $x_6(t)$ and $x_7(t)$ (i.e. the levels of water in the tanks CO and TA respectively). In other words, the *semantic* to the functions f_2 and f_3 of the functional model $SM(\Omega)$ is the following:

$$x_7(t) = f_2(x_5(t)) = (1 - \phi_8) \cdot \int x_5(t)dt \tag{4.19}$$

$$x_6(t) = f_3(x_5(t)) = \phi_8 \cdot \int x_5(t)dt \tag{4.20}$$

As a consequence, the functions f_2 and f_3 must be re-defined under the algebraic form of $x_7 = f_2(x_5, \phi_8)$ and $x_6 = f_3(x_5, \phi_8)$ (cf. the tables 4.13 and 4.15).

Let us apply firstly the *FSM transformation* to the function f_2. Let us denote ϕ_2 the internal variable of f_2 and Δ_{ϕ_2} its definition domain. The output of the f_2 function can have three values: δ_{70} (*zero*), δ_{71} (*low*) and δ_{72} (*normal*). So let us define also three values for ϕ_2: δ_{20} denoting the fact that the tank TA is empty, δ_{21} denoting the fact that the tank TA contains few water, and δ_{22} denoting the fact the tank is normally full (cf. table 4.12).

Δ_{ϕ_2}	
Constants	*Interpretation*
δ_{20}	*Empty tank*
δ_{21}	*Few water in the tank*
δ_{22}	*Tank normally full*

Table 4.12: Definition Domain Δ_{ϕ_2} of ϕ_2

The table 4.13 constitutes then an algebraic representation of the function f_2. Again, this table specifies an automata that will be described in the TOM4D behavioral model.

$f_2 : \Delta_{x_5} \to \Delta_{x_7}$			
ϕ_2	x_5	$f_2(x_5)$	*Interpretation*
δ_{20}	δ_{50}	δ_{70}	*Empty tank, no input flow ⇒ null level*
δ_{20}	δ_{51}	δ_{71}	*Empty tank, low flow ⇒ low level*
δ_{20}	δ_{52}	δ_{71}	*Empty tank, normal flow ⇒ low level*
δ_{21}	δ_{50}	δ_{71}	*Few water in the tank, no input flow ⇒ low level*
δ_{21}	δ_{51}	δ_{72}	*Few water in the tank, a low input flow ⇒ normal level*
δ_{21}	δ_{52}	δ_{72}	*Few water in the tank, a low input flow ⇒ normal level*
δ_{22}	-	δ_{72}	*Tank normally full ⇒ normal level*

Table 4.13: Definition of the Generic Function f_2

Similarly, let us denote ϕ_3 the internal variable of f_3 and $Delta_{\phi_3}$ its definition domain. Because the output value can have two values δ_{60} (*presence of water*) and δ_{61} (*absence of water*), ϕ_3 must also have two values: δ_{31} denoting the fact that the tank CO contains water, and δ_{30} denoting the fact the tank is empty (cf. table 4.14).

Δ_{ϕ_3}	
Constants	*Interpretation*
δ_{30}	*Empty tank*
δ_{31}	*The tank contains water*

Table 4.14: Definition Domain Δ_{ϕ_3} of ϕ_3

$f_3 : \Delta_{x_5} \times \Delta_{\phi_8} \to \Delta_{x_6}$				
ϕ_3	x_5	ϕ_8	$f_3(x_5, \phi_8)$	*Interpretation*
δ_{30}	-	δ_{81}	δ_{61}	*Empty tank,* $\phi_8 = 0 \Rightarrow$ *absence of water*
δ_{30}	δ_{50}	-	δ_{61}	*Empty tank, no input flow* $\Rightarrow$ *absence of water*
δ_{30}	δ_{51}	δ_{80}	δ_{60}	*Empty tank, low input flow,* $\phi_8 > 0 \Rightarrow$ *presence of water*
δ_{30}	δ_{52}	δ_{80}	δ_{60}	*Empty tank, normal input flow,* $\phi_8 > 0 \Rightarrow$ *presence of water*
δ_{31}	-	-	δ_{60}	*Not empty tank* $\Rightarrow$ *presence of water*

Table 4.15: Definition of the Generic Function f_3

The table 4.15 constitutes then an algebraic representation of the function f_3.

Recalling that the variables ϕ_2 and ϕ_3 are internal variables of the functions f_2 and f_3 respectively, the analysis with the hydraulic ToS of the precedeing step alows to update the functional model $FM(\Omega)$ (cf. figure 4.15).

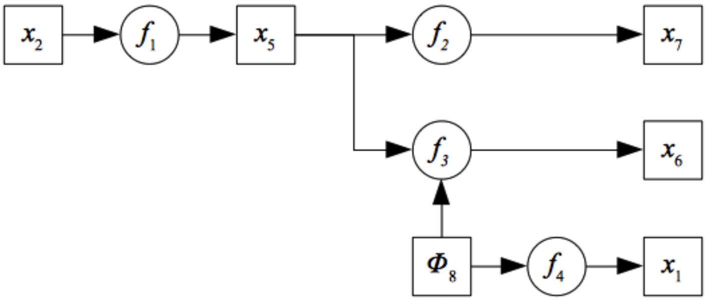


Figure 4.15: Generic Functional Model $FM(X(t))$

Formaly, the Generic Functional Model $FM(X(t))$ of the hydraulic system is given in the definition 4.7.

Definition 4.7 $FM(X(t))$
The Functional Model $FM(X(t))$ *of the hydraulic process* $X(t)$ *is the structure* $< \Delta, F, R^f >$ *where,*

- $\Delta = \Delta_{x_1} \cup \Delta_{x_2} \cup \Delta_{x_5} \cup \Delta_{x_6} \cup \Delta_{x_7} \cup \Delta_{\phi_8}$.
- $F = \{f_1 : \Delta^{x_1} \times \Delta^{x_5}, f_2 : \Delta^{x_5} \times \Delta^{x_7}, f_3 : \Delta^{x_5} \times \Delta^{\phi_8} \times \Delta^{x_7}, f_4 : \Delta^{\phi_8} \times \Delta^{x_1}\}$.
- $R^f = \{x_5 = f_1(x_2), x_7 = f_2(x_5), x_6 = f_3(x_5, \phi_8), x_1 = f_4(\phi_8)\}$.

The definition domain Δ_{x_i} of the variables are given in the table 4.6 for x_1, x_2, x_5, x_6 and x_7 and 4.10 for ϕ_8. The function f_1, f_2, f_3 and f_4 are specified in the tables 4.9, 4.13, 4.15 and 4.11 respectively.

The generic structural model $SM(X(t))$ can now be defined. The analysis of the hydraulic system leads to define a new component, denoted C_8, that merges the components c_2 (the *pump*

P), c_1 (the *pipe* PI). The analysis have nothing to say about the sensors c_3 (the sensor $flow_p$), c_7 ($level_{TA}$) and c_6 ($level_{TA}$), they are considered as always available. They have then been merged in their respective components c_8, c_7 and c_8. This leads to a simplified the generic structural model graphically represented in figure 4.16.

Definition 4.8 $SM(X(t))$
The Generic Structural Model $SM(X(t))$ *is a structure* $< COMPS, R^p, R^x >$ *where:*

- $COMPS = \{c_0, c_8, c_9\}$
- $R^p = \{out_1(c_8) = in(c_9), out_1(c_8) = in_1(c_0), out_2(c_8) = in_2(c_0)\}$
- $R^x = \{x_1 = out_3(c_8), x_2 = out_4(c_8), x_5 = out_1(c_8), \phi_8 = out_2(c_8), x_6 = out(c_0), x_7 = out(c_9)\}$

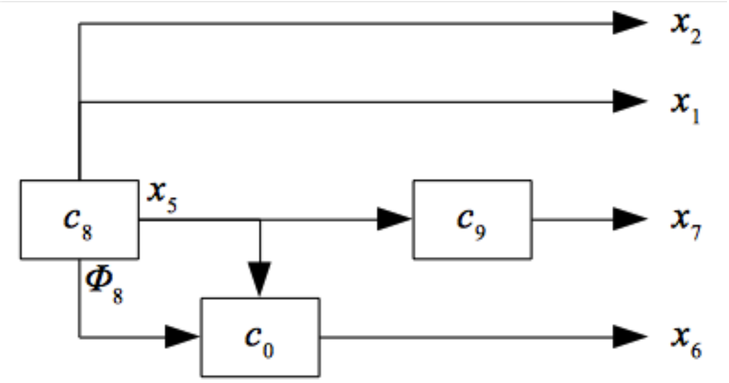


Figure 4.16: Generic Structural Model $SM(X(t))$

This generic structural model defines the hydraulic process $X(t)$ as a network of dynamic process $X(t) = X_8(t) \cup X_7(t) \cup X_6(t)$ where:

- $X_8(t) = \{x_1(t), x_2(t), x_5(t), \phi_8(t)\}$
- $X_7(t) = \{x_5(t), x_7(t)\}$
- $X_6(t) = \{x_5(t), x_6(t), \phi_8(t)\}$

This means that:

- $X_8(t)$ implements the functions f_1 and f_4.
- $X_7(t)$ implements the function f_2.
- $X_6(t)$ implements the function f_3.

The interest of the network structure is the simplification of the building of the generic behavioral model : the functions f_1 and f_4 being purely algebraic, the process $X_8(t)$ does not requires a behavioral model. Inversely, the processes $X_7(t)$ and $X_6(t)$ requires a behavioral model because the functions f_2 and f_3 specifies an automata. Finally, the propositions contained in the perception model $PM(X(t))$ will be distributed over the different processes.

4.5.3.1 Step 3.1: Generic Modeling of $X_8(t)$

The Perception Model of the process $X_8(t)$ is given in the definition 4.9:

Definition 4.9 $PM(X_8(t))$
The Perception Model $PM(X_8(t))$ of the process $X_8(t) = \{x_1(t),\ x_2(t),\ x_5(t),\ \phi_8(t)\}$ is the structure $< X_8, \Psi_8, R^{q_8} >$ where:

- $X_8 = \{x_1, x_2, x_5, \phi_8\}$,
- $\Psi = \{\Psi_{51}, \Psi_{52}, \Psi_8\}$
- $R^{q_8} = R^{goal_8} \cup R^{n_8} \cup R^{ab_8}$.
  - $R^{goal_8} = \Phi$.
  - $R^{n_8} = \{\forall t \geq t_0, x_1(t) = \delta_{11} \wedge x_2(t) = \delta_{22}\}$.
  - $R^{ab_8} = \{\forall t \geq t_0, x_1(t) = \delta_{10} \vee x_2(t) = \delta_{20} \vee x_2(t) = \delta_{21}\}$.

The main goal of $X(t)$ being concerned with $X_7(t)$, no goal is assigned to $X_8(t)$. The normal operation R^{n_8} means that the *mode of behavior* of the *pump P* and the *pipe PI* must be *ok*: the output flow of the *pump P* is equal to the input flow of the *pipe PI* and the input and output flows of the *pipe PI* are equal. In that case, all the water provided by the *pump P* goes into the *tank TA*. As a consequence, the abnormal operations R^{ab_8} is the complement of R^{n_8}: the *pipe PI* is *leaking* (i.e. $x_1(t) = \delta_{10}$) or the *pump P* is either *blocked* ($x_2(t) = \delta_{20}$) or *leaking* ($x_2(t) = \delta_{21}$).

The structural model of $X_8(t)$ is given in the definition 4.10 and is represented in the figure 4.17.

Definition 4.10 $SM(X_8(t))$
The structural model $SM(X_8(t))$ is the structure $< COMPS_8, R^{p_8}, R^{x_8} >$ where:

- $COMPS_8 = \{c_1, c_2, c_5\}$
- $R^{p_8} = \{out_1(c_2) = in_1(c_1), out_2(c_2) = in(c_5)\}$
- $R^{x_8} = \{x_1 = out_2(c_1), x_2 = out_1(c_2), x_5 = out(c_5), \phi_8 = out_2(c_1)\}$

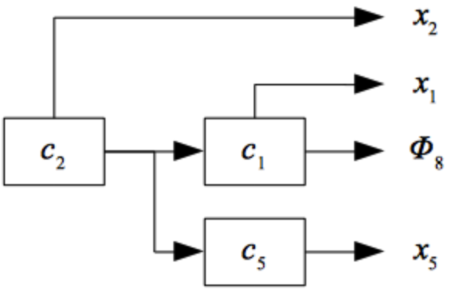


Figure 4.17: Generic Structural Model $SM(X_8(t))$

The Generic Functional Model $FM(X_8(t))$ of $X_8(t)$ is given in the definition 4.11 (cf. figure 4.18.

Definition 4.11 $FM(X_8(t))$
The Functional Model $FM(X_8(t))$ of the hydraulic process $X_8(t)$ is the structure $< \Delta_8, F_8, R^{f_8} >$ where,

- $\Delta_8 = \Delta_{x_1} \cup \Delta_{x_2} \cup \Delta_{x_5} \cup \Delta_{\phi_8}$.
- $F_8 = \{f_1 : \Delta^{x_1} \times \Delta^{x_5}, f_4 : \Delta^{\phi_8} \times \Delta^{x_1}\}$.
- $R^f = \{x_5 = f_1(x_2), x_1 = f_4(\phi_8)\}$.

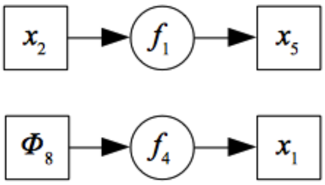


Figure 4.18: Generic Functional Model $FM(X_8(t))$

The specification of the functions f_1 and f_4 is recalled in the table 4.16.

$f_1 : \Delta_{x_2} \to \Delta_{x_5}$		
x_2	f_1	*Interpretation:* $x_2 = f_1(x_5)$
δ_{20}	δ_{50}	$x_2 = \delta_{20} \Rightarrow x_5 = \delta_{50}$
δ_{21}	δ_{51}	$x_2 = \delta_{21} \Rightarrow x_5 = \delta_{51}$
δ_{22}	δ_{52}	$x_2 = \delta_{22} \Rightarrow x_5 = \delta_{52}$

$f_4 : \Delta_{\phi_8} \to \Delta_{x_1}$		
ϕ_8	f_4	*Interpretation:* $x_1 = f_4(\phi_8)$
δ_{80}	δ_{10}	$\phi_8 > 0 \Rightarrow x_1 = \delta_{10}$
δ_{81}	δ_{11}	$\phi_8 = 0 \Rightarrow x_1 = \delta_{10}$

Table 4.16: Definition of the Generic Functions f_1 and f_4

The process $X_8(t)$ being a purely combinatorial machine, the *algebraic to automata transformation* is required to build its behavioral model $BM(X_8(t))$ (cf. definition 4.12).

Definition 4.12 $BM(X_8(t))$
The behavior model $BM(X_8(t))$ of the dynamic process $X_8(t) = \{x_1(t),\ x_2(t),\ x_5(t),\ \phi_8(t)\}$ is the structure $< S_8, C_8, R^{s_8} >$ where:

- *The set $S_8 = \{$*
  $s_{80} = \{r_{801}(C_{20}, C_{50})\}$, $s_{81} = \{r_{811}(C_{21}, C_{51})\}$, $s_{82} = \{r_{820}(C_{22}, C_{52})\}$
  $s_{83} = \{r_{831}(C_{80}, C_{10})\}$, $s_{84} = \{r_{841}(C_{81}, C_{11})\}$
  $\}$ of 5 discernible states definig 5 sequential binary relations.
- *The set $C_8 = \{$*
  $C_{20} = \{(x_2, \delta_{20})\}, C_{21} = \{(x_2, \delta_{21})\}, C_{22} = \{(x_2, \delta_{22})\}$,
  $C_{50} = \{(x_5, \delta_{50})\}, C_{51} = \{(x_5, \delta_{51})\}, C_{52} = \{(x_5, \delta_{52}),\}$
  $C_{80} = \{(\phi_8, \delta_{80})\}, C_{81} = \{(\phi_8, \delta_{81})\}$,
  $C_{10} = \{(x_{10}, \delta_{10})\}, C_{11} = \{(x_{10}, \delta_{11})\}$
  $\}$ made of 6 observations classes.
- *The empty set $R^{s_8} = \{\phi\}$.*

Figure 4.19 proposes a graphical representation of $BM(X_8(t))$.

So, finally, the model $M(X_8(t))$ of the dynamic process $X_8(t) = \{x_1(t),\ x_2(t),\ x_5(t),\ \phi_8(t)\}$ is the tuple $M(X_8(t)) =< PM(X_8(t)),\ SM(X_8(t)),\ FM(X_8(t)),\ \Phi >$.

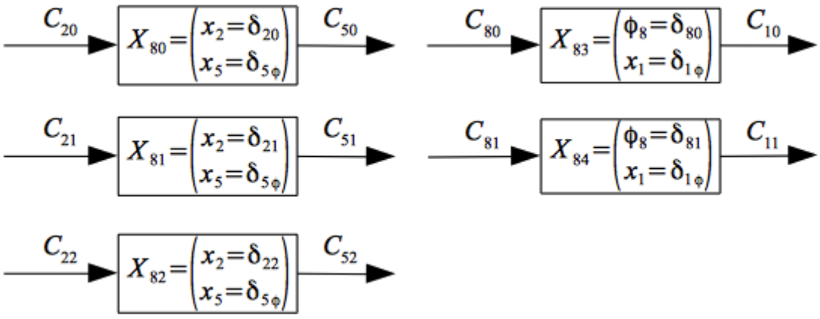

C_{20}
$X_{80}=\begin{pmatrix}x_2=\delta_{20}\\x_5=\delta_{5\phi}\end{pmatrix}$
C_{50}
C_{80}
$X_{83}=\begin{pmatrix}\phi_8=\delta_{80}\\x_1=\delta_{1\phi}\end{pmatrix}$
C_{10}
C_{21}
$X_{81}=\begin{pmatrix}x_2=\delta_{21}\\x_5=\delta_{5\phi}\end{pmatrix}$
C_{51}
C_{81}
$X_{84}=\begin{pmatrix}\phi_8=\delta_{81}\\x_1=\delta_{1\phi}\end{pmatrix}$
C_{11}
C_{22}
$X_{82}=\begin{pmatrix}x_2=\delta_{22}\\x_5=\delta_{5\phi}\end{pmatrix}$
C_{52}

Figure 4.19: Generic Behavioral Model $BM(X_8(t))$

4.5.3.2 Step 3.2: Generic Modeling of $X_7(t)$

The Perception Model of the process $X_7(t)$ is given in the definition 4.13:

Definition 4.13 $PM(X_7(t))$
The Perception Model $PM(X_7(t))$ of the process $X_7(t) = \{x_5(t), x_7(t)\}$ is the structure $< X_7, \Psi_7, R^{q_7} >$ where:

- $X_7 = \{x_5, x_7\}$,
- $\Psi_7 = \{\Psi_{71}, \Psi_{72}\}$
- $R^{q_7} = R^{goal_7} \cup R^{n_7} \cup R^{ab_7}$.
  - $R^{goal_7} = \{\exists t_i, t_i \geq t_0, \forall t \geq t_i, x_7(t) \geq \Psi_{72}\}$
  - $R^{n_7} = \{\forall t \geq t_i, x_7(t) \geq \Psi_{72}\}$
  - $R^{ab_7} = \{\forall t \geq t_i, x_7(t) < \Psi_{72}\}$

The aim of the operations of $X_7(t)$ is to maintain the level of the water in the *tank TA* at its normal level (i.e. $level_{TA} = nrm_{TA}$). The time-stamp t_i denotes the time at which the level of the *tank TA* reaches for the first time its normal level.

The structural model $SM(X_7(t))$ of $X_7(t)$ is given in the definition 4.14 and is represented in the figure 4.20.

Definition 4.14 $SM(X_7(t))$
The structural model $SM(X_7(t))$ is the structure $< COMPS_7, R^{p_7}, R^{x_7} >$ where:

- $COMPS_7 = \{c_3, c_7\}$
- $R^{p_7} = \{out(c_3) = in(c_7)\}$
- $R^{x_7} = \{x_7 = out(c_7)\}$

The Generic Functional Model $FM(X_7(t))$ of the hydraulic system is given in the definition 4.15:

Definition 4.15 $FM(X_7(t))$
The Functional Model $FM(X_7(t))$ of the process $X(t_7)$ is the structure $< \Delta_7, F_7, R^{f^7} >$ where:

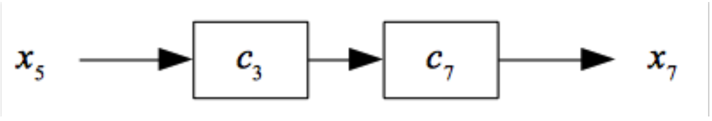


Figure 4.20: Generic Structural Model $SM(X_7(t))$

- $\Delta_7 = \Delta_{x_5} \cup \Delta_{x_7}$.
- $F_7 = \{f_2 : \Delta^{x_5} \times \Delta^{x_7}\}$.
- $R^{f_7} = \{x_7 = f_2(x_5)\}$.

The specification of the function f_2 of the equation $x_7(t) = f_2(x_5(t))$ is recalled in the table 4.17. This function containing an integrator, the rules of $x_7(t) = f_2(x_5(t))$ must be implemented in the behavioral model $BM(X_7(t))$ of $X_7(t)$.

$f_2 : \Delta_{x_5} \to \Delta_{x_7}$				
ϕ_2	x_5	f_2	*Interpretation:* $x_7 = f_2(x_5)$	*Physical Interpretation*
δ_{22}	δ_{50}	δ_{70}	$\phi_2 = \delta_{22} \wedge x_5 = \delta_{50} \Rightarrow x_7 = \delta_{70}$	*Empty,* $flow_p = zro_p \to level_{TA} = zro_{TA}$
δ_{20}	δ_{51}	δ_{71}	$\phi_2 = \delta_{20} \wedge x_5 = \delta_{51} \Rightarrow x_7 = \delta_{71}$	*Empty,* $flow_p = low_p \to level_{TA} = low_{TA}$
δ_{20}	δ_{52}	δ_{71}	$\phi_2 = \delta_{20} \wedge x_5 = \delta_{52} \Rightarrow x_7 = \delta_{71}$	*Empty,* $flow_p = nrm_p \to level_{TA} = low_{TA}$
δ_{21}	δ_{50}	δ_{71}	$\phi_2 = \delta_{21} \wedge x_5 = \delta_{50} \Rightarrow x_7 = \delta_{71}$	*Few,* $flow_p = zro_p \to level_{TA} = low_{TA}$
δ_{21}	δ_{51}	δ_{72}	$\phi_2 = \delta_{21} \wedge x_5 = \delta_{51} \Rightarrow x_7 = \delta_{71}$	*Few,* $flow_p = low_p \to level_{TA} = nrm_{TA}$
δ_{21}	δ_{52}	δ_{72}	$\phi_2 = \delta_{21} \wedge x_5 = \delta_{52} \Rightarrow x_7 = \delta_{72}$	*Few,* $flow_p = nrm_p \to level_{TA} = nrm_{TA}$
δ_{22}	-	δ_{72}	$\phi_2 = \delta_{22} \Rightarrow x_7 = \delta_{72}$	*Water* $\to level_{TA} = nrm_{TA}$

Table 4.17: Definition of the Generic Function f_2 of $X_7(t)$

To $BM(X_7(t))$ the behavioral model, let us make the followings hypothesis:

- H1: The set $\Delta_7 = \{\delta_{70}, \delta_{71}, \delta_{72}\}$ is *ordinal.* This means that the values of x_7 increases (or decreases) to the next (previous) value in Δ_7. For example, if $x_7(t_k) = \delta_{70}$, x_7 can only increases to the value δ_{71}.
- H2: The normal value (i.e. nrm_p) of the water flow $flow_p$ in the couple (*pump*, *pipe*) is always sufficient to provide water in c_7, the (*tank TA*). This means that even when $\phi_8 > 0$ (i.e. the *pipe* is *leaking*), the water flow is still sufficient to increase the level in the *tank TA*.

These two hypotheses and the table 4.17 leads to the behavioral model of the definition 4.16:

Definition 4.16 $BM(X_7(t))$
The behavior model $BM(X_7(t))$ *of the dynamic process*

$X_7(t) = \{\ x_5(t),\ x_7(t)\}$ *is the structure* $< S_7, C_7, R^{s_7} >$ *where:*

- *The set* $S_7 = \{$
  $s_{70} = \{r_{701}(C_{50}, C_{51}), r_{702}(C_{70}, C_{51})\}$,
  $s_{71} = \{r_{711}(C_{51}, C_{50}), r_{712}(C_{51}, C_{52}), r_{713}(C_{51}, C_{71}), r_{714}(C_{70}, C_{50}), r_{715}(C_{70}, C_{52}), r_{716}(C_{70}, C_{71})\}$,
  $s_{72} = \{r_{721}(C_{51}, C_{50}), r_{722}(C_{51}, C_{52}), r_{723}(C_{51}, C_{70}), r_{724}(C_{51}, C_{72})$,

$r_{725}(C_{71}, C_{50}), r_{726}(C_{71}, C_{52}), r_{727}(C_{71}, C_{70}), r_{728}(C_{71}, C_{72})\},$
$s_{73} = \{r_{731}(C_{52}, C_{71}), r_{732}(C_{52}, C_{51})\},$
$s_{74} = \{r_{741}(C_{50}, C_{51}), r_{742}(C_{50}, C_{70}), r_{743}(C_{71}, C_{51}), r_{744}(C_{71}, C_{70})\},$
$s_{75} = \{r_{751}(C_{51}, C_{50)}, r_{752}(C_{51}, C_{52}), r_{753}(C_{51}, C_{71}), r_{754}(C_{72}, C_{50}), r_{755}(C_{72}, C_{52}), r_{756}(C_{72}, C_{71})\},$
$s_{76} = \{r_{761}(C_{52}, C_{51}), r_{762}(C_{52}, C_{72}), r_{763}(C_{71}, C_{51}), r_{764}(C_{71}, C_{72})\},$
$s_{77} = \{r_{771}(C_{50}, C_{51}), r_{772}(C_{50}, C_{71})\},$
$s_{78} = \{r_{781}(C_{52}, C_{51}), r_{782}(C_{72}, C_{51})\}$
$\}$ *of 9 discernible states defining 36 sequential binary relations.*

- *The set* $C_7 = \{$
  $C_{50} = \{(x_5, \delta_{50})\}, C_{51} = \{(x_5, \delta_{51})\}, C_{52} = \{(x_5, \delta_{52})\},$
  $C_{70} = \{(x_7, \delta_{70})\}, C_{71} = \{(x_7, \delta_{71})\}, C_{72} = \{(x_7, \delta_{72})\}$
  $\}$ *of 6 observations classes.*

- *The set* $R^{s_7} = \{$
  $s_{70} = \gamma(s_{71}, C_{50}),\ s_{70} = \gamma(s_{74}, C_{70}),$
  $s_{71} = \gamma(s_{70}, C_{51}),\ s_{71} = \gamma(s_{73}, C_{51}),\ s_{71} = \gamma(s_{72}, C_{70}),$
  $s_{73} = \gamma(s_{71}, C_{52}),$
  $s_{74} = \gamma(s_{72}, C_{50}),\ s_{74} = \gamma(s_{77}, C_{71}),$
  $s_{72} = \gamma(s_{71}, C_{71}),\ s_{72} = \gamma(s_{76}, C_{51}),\ s_{72} = \gamma(s_{75}, C_{71}),\ s_{72} = \gamma(s_{74}, C_{51}),$
  $s_{76} = \gamma(s_{72}, C_{52}),\ s_{76} = \gamma(s_{73}, C_{71}),$
  $s_{77} = \gamma(s_{75}, C_{50}),$
  $s_{75} = \gamma(s_{72}, C_{72}),\ s_{75} = \gamma(s_{78}, C_{51}),\ s_{75} = \gamma(s_{77}, C_{51}),$
  $s_{78} = \gamma(s_{76}, C_{72}),\ s_{78} = \gamma(s_{75}, C_{52})$
  $\}$ *of 19 predicates.*

The figure 4.21 provide a graphical representation of $BM(X_7(t))$ where each state s_k has been replaced with its corresponding *discernible state vector* X_k, defined on $\Delta_5 \times \Delta_7$, to facilitate the reading of the model. In this figure, the boxes represent the discernible states of the process $X_7(t)$ and the relations from the left to the right are over the arrows, the relation that are under the arrows going from right to the left. According to the TOT, a discernible state s_i is defined with a set of sequential binary relation between two observation classes (i.e. $r(C_i, C_j)$), and each discernible state can be labelled with a particular value. In the figure 4.21, the discernible states have been labeled with the values X_{7i}, $i = 0...8$, of a two dimensions vector, x_5 and x_6, so that each label X_{7i} corresponds to a particular state s_{7i}. For example, the state $X_{70} = \{x_5 = \delta_{50}, x_7 = \delta_{70}\}$ corresponds to the *initial state* representing the fact that the water flow in the pipe is null and the level in the tank TA is zero.

The model $M(X_7(t))$ of the dynamic process $X_7 = \{x_5, x_7\}$ is then the tuple $M(X_7(t)) =< PM(X_7(t)), SM(X_7(t)), FM(X_7(t)), BM(X_7(t)) >$.

It is to note that this model is completely *symmetric* according the values of x_5: each state where a class C_{50} (respectively C_{51}) allows to enter contain the symmetric class C_{51} (respectively C_{50}) as an output class. The symmetry is less strong with the values of x_7: for example, there is no way to go back to the state s_{73} with a class C_{70} from the state x_{76}. This is the behavioral translation of the fact that when the $flow_p$ is nrm_p (i.e. $x_5 = \delta_{52}$), it has been supposed that $flow_p$ is *necessary and sufficient* to increase the water level in the *tank TA* (i.e. $level_{TA}$ and

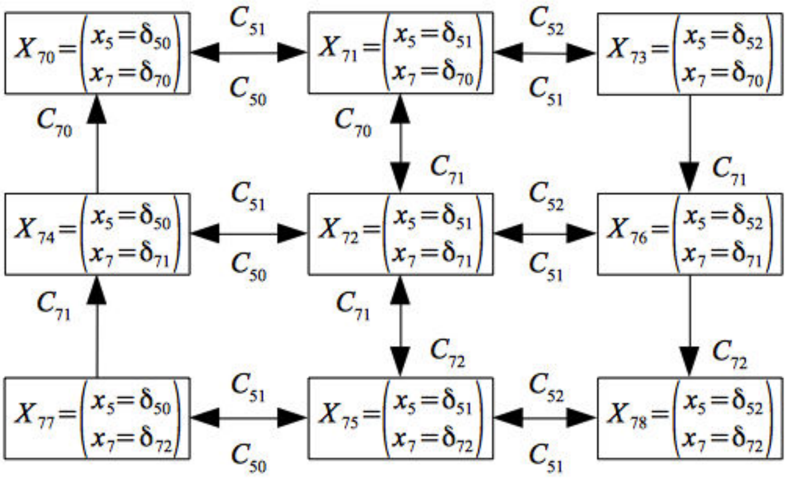

$X_{70}=\begin{pmatrix}x_5=\delta_{50}\\x_7=\delta_{70}\end{pmatrix}$
C_{51}
C_{50}
$X_{71}=\begin{pmatrix}x_5=\delta_{51}\\x_7=\delta_{70}\end{pmatrix}$
C_{52}
C_{51}
$X_{73}=\begin{pmatrix}x_5=\delta_{52}\\x_7=\delta_{70}\end{pmatrix}$
C_{70}
C_{70}
C_{71}
C_{71}
$X_{74}=\begin{pmatrix}x_5=\delta_{50}\\x_7=\delta_{71}\end{pmatrix}$
C_{51}
C_{50}
$X_{72}=\begin{pmatrix}x_5=\delta_{51}\\x_7=\delta_{71}\end{pmatrix}$
C_{52}
C_{51}
$X_{76}=\begin{pmatrix}x_5=\delta_{52}\\x_7=\delta_{71}\end{pmatrix}$
C_{71}
C_{71}
C_{72}
C_{72}
$X_{77}=\begin{pmatrix}x_5=\delta_{50}\\x_7=\delta_{72}\end{pmatrix}$
C_{51}
C_{50}
$X_{75}=\begin{pmatrix}x_5=\delta_{51}\\x_7=\delta_{72}\end{pmatrix}$
C_{52}
C_{51}
$X_{78}=\begin{pmatrix}x_5=\delta_{52}\\x_7=\delta_{72}\end{pmatrix}$

Figure 4.21: Generic Behavioral Model $BM(X_7(t))$

maintain it at its higher level δ_{72}. Such a property can not be deduced from the logical properties of the structure or the functions of a dynamic process: it must be translated in its behavioral model.

This allows to illustrate two important points of the TOM4D methodology:

- The behavioral model of TOM4D aims at formalizing such pieces of knowledge concerning some particular properties about the behavior of a dynamic process.
- These piece of knowledge *breaks* a part of the symetry in the behavioral model of the process. This brings to light the *specificities* of the process, what makes it unique.

In practice, the introduction of such pieces of knowledge leads to *prune* the completely symmetric behavioral model: this constitutes an excellent way to model the behavior of a process. In other words, the particular properties about the behavior of a dynamic process leads to the *simplification* of its behavioral model. So, the TOM4D methodology *recommends* to firstly build a fully symmetric behavioral model, and next to prune it with the progressive introduction of its specificities.

4.5.3.3 Step 3.3: Generic Modeling of $X_6(t)$

The Perception Model of the process $X_6(t)$ is given in the definition 4.17:

Definition 4.17 $PM(X_6(t))$
The Perception Model $PM(X_6(t))$ *of the process* $X_6(t) = \{x_5(t), x_6(t), \phi_8\}$ *is the structure* $< X_6, \Psi_6, R^{q_6} >$ *where:*

- $X_6 = \{x_5, x_6, \phi_8\}$,
- $\Psi_6 = \{\Psi_{60}\}$
- $R^{q_6} = R^{goal_6} \cup R^{n_6} \cup R^{ab_6}$.
  - $R^{goal_6} = \Phi$

- $R^{n_6} = \{\forall t \geq t_0, x_6(t) \leq \Psi_{60}\}$
- $R^{ab_6} = \{\forall t \geq t_0, x_6(t) > \Psi_{60}\}$

The aim of the operations of $X_6(t)$ is simply to maintain empty the *tank CO* (i.e. $level_{CO} = abs_{CO}$).

The structural model $SM(X_6(t))$ of $X_6(t)$ is given in the definition 4.18 and is represented in the figure 4.22.

Definition 4.18 $SM(X_6(t))$
The structural model $SM(X_6(t))$ *is the structure* $< COMPS_6, R^{p_6}, R^{x_6} >$ *where:*

- $COMPS_6 = \{c_4, c_6\}$
- $R^{p_6} = \{out(c_4) = in(c_6)\}$
- $R^{x_6} = \{in_1(c_4) = x_5, in_2(c_4) = \phi_8, x_6 = out(c_6)\}$

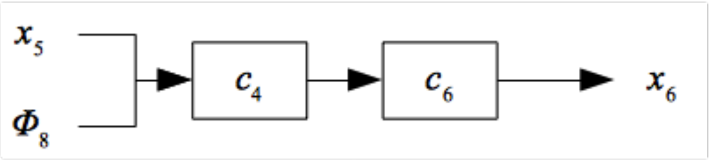


Figure 4.22: Generic Structural Model $SM(X_6(t))$

The Generic Functional Model $FM(X_6(t))$ of the hydraulic system is given in the definition 4.19:

Definition 4.19 $FM(X_6(t))$
The Functional Model $FM(X_6(t))$ *of the process* $X(t_6)$ *is the structure* $< \Delta_6, F_6, R^{f_6} >$ *where:*

- $\Delta_6 = \Delta_{x_5} \cup \Delta_{\phi_8} \cup \Delta_{x_6}$.
- $F_6 = \{f_3 : \Delta_{x_5} \times \Delta_{\phi_8} \times \Delta_{x_6}\}$.
- $R^{f_6} = \{x_6 = f_3(x_5, \phi_8)\}$.

The table 4.18 recalls the specification of the function f_3.

$f_3 : \Delta_{x_5} \times \Delta_{\phi_8} \rightarrow \Delta_{x_6}$					
ϕ_3	x_5	ϕ_8	f_3	*Interpretation:* $x_6 = f_3(x_5, \phi_8)$	*Physical Interpretation:*
δ_{30}	-	δ_{81}	δ_{61}	$\phi_3 = \delta_{30} \wedge \phi_8 = \delta_{81} \Rightarrow x_6 = \delta_{61}$	*E*, $\phi_8 = 0 \rightarrow abs_{CO}$
δ_{30}	δ_{50}	-	δ_{61}	$\phi_3 = \delta_{30} \wedge x_5 = \delta_{50} \Rightarrow x_6 = \delta_{61}$	*E*, $zro_p \rightarrow abs_{CO}$
δ_{30}	δ_{51}	δ_{80}	δ_{60}	$\phi_3 = \delta_{30} \wedge x_5 = \delta_{51} \wedge \phi_8 = \delta_{80} \Rightarrow x_6 = \delta_{60}$	*E*, low_p, $\phi_8 > 0 \rightarrow pre_{CO}$
δ_{30}	δ_{52}	δ_{80}	δ_{60}	$\phi_3 = \delta_{30} \wedge x_5 = \delta_{52} \wedge \phi_8 = \delta_{80} \Rightarrow x_6 = \delta_{60}$	*E*, nrm_p, $\phi_8 > 0 \rightarrow pre_{CO}$
δ_{31}	-	-	δ_{60}	$\phi_3 = \delta_{31} \Rightarrow x_6 = \delta_{60}$	*Water* $\rightarrow pre_{CO}$

Table 4.18: Definition of the Generic Function f_3 of $X_6(t)$ (where *E* denotes *Empty*)

The function f_3 containing also an integrator, the rules of $x_6 = f_3(x_5, \phi_8)$ will be implemented in the behavioral model of the definition 4.20:

Definition 4.20 $BM(X_6(t))$
The behavior model $BM(X_6(t))$ *of* $X_6(t) = \{x_5(t), x_6(t), \phi_8\}$ *is the tuple* $< S_6, C_6, R^{s_6} >$ *where:*

- *The set* $S_6 = \{$
  $s_{60} = \{r_{601}(C_{50}, C_{51}), r_{602}(C_{50}, C_{80}), r_{603}(C_{81}, C_{51}), r_{604}(C_{81}, C_{80})\},$
  $s_{61} = \{r_{611}(C_{50}, C_{51}), r_{612}(C_{50}, C_{81}), r_{613}(C_{80}, C_{51}), r_{614}(C_{80}, C_{81})\},$
  $s_{62} = \{r_{621}(C_{51}, C_{50}), r_{622}(C_{51}, C_{52}), r_{623}(C_{51}, C_{60}), r_{624}(C_{80}, C_{50}), r_{625}(C_{80}, C_{52}), r_{626}(C_{80}, C_{60})\},$
  $s_{63} = \{r_{631}(C_{52}, C_{51}), r_{632}(C_{52}, C_{60}), r_{633}(C_{80}, C_{51}), r_{634}(C_{80}, C_{60})\},$
  $s_{64} = \{r_{641}(C_{60}, C_{52}), r_{642}(C_{60}, C_{81}), r_{643}(C_{51}, C_{52}), r_{644}(C_{51}, C_{81})\},$
  $s_{65} = \{r_{651}(C_{52}, C_{51}), r_{652}(C_{52}, C_{81}), r_{653}(C_{60}, C_{51)}, r_{654}(C_{60}, C_{81})\},$
  $s_{66} = \{r_{661}(C_{51}, C_{52}), r_{662}(C_{51}, C_{61}), r_{663}(C_{81}, C_{52}), r_{664}(C_{81}, C_{61})\},$
  $s_{67} = \{r_{671}(C_{52}, C_{51}), r_{672}(C_{52}, C_{61}), r_{673}(C_{81}, C_{51)}, r_{674}(C_{81}, C_{61})\},$
  $s_{68} = \{r_{681}(C_{51}, C_{50}), r_{682}(C_{51}, C_{52}), r_{683}(C_{51}, C_{80}), r_{684}(C_{61}, C_{50}), r_{685}(C_{61}, C_{52}), r_{686}(C_{61}, C_{80})\},$
  $s_{69} = \{r_{691}(C_{52}, C_{51}), r_{692}(C_{52}, C_{80}), r_{693}(C_{61}, C_{51}), r_{694}(C_{61}, C_{80})\}$
  $\}$ *of 10 discernible states made with 44 sequential binary relations.*

- *The set* $C_6 = \{$
  $C_{50} = \{(x_5, \delta_{50})\}, C_{51} = \{(x_5, \delta_{51})\}, C_{52} = \{(x_5, \delta_{52})\},$
  $C_{60} = \{(x_6, \delta_{60})\}, C_{61} = \{(x_6, \delta_{61})\}, C_{62} = \{(x_6, \delta_{62})\},$
  $C_{80} = \{(\phi_8, \delta_{80})\}, C_{81} = \{(\phi_8, \delta_{81})\}$
  $\}$ *of 8 observations classes.*

- *The set* $R^{s_6} = \{$
  $s_{60} = \gamma(s_{61}, C_{81}),\ s_{60} = \gamma(s_{68}, C_{50}),$
  $s_{61} = \gamma(s_{60}, C_{80}),\ s_{61} = \gamma(s_{62}, C_{50}),$
  $s_{62} = \gamma(s_{61}, C_{51}),\ s_{62} = \gamma(s_{63}, C_{51}),\ s_{62} = \gamma(s_{68}, C_{80}),$
  $s_{63} = \gamma(s_{62}, C_{52}),\ s_{63} = \gamma(s_{69}, C_{80}),$
  $s_{64} = \gamma(s_{62}, C_{60}),\ s_{64} = \gamma(s_{65}, C_{51}),$
  $s_{65} = \gamma(s_{63}, C_{60}),\ s_{65} = \gamma(s_{64}, C_{52}),$
  $s_{66} = \gamma(s_{64}, C_{81}),\ s_{66} = \gamma(s_{67}, C_{51}),$
  $s_{67} = \gamma(s_{65}, C_{81}),\ s_{67} = \gamma(s_{66}, C_{52}),$
  $s_{68} = \gamma(s_{66}, C_{61}),\ s_{68} = \gamma(s_{69}, C_{51}),\ s_{68} = \gamma(s_{60}, C_{51}),$
  $s_{69} = \gamma(s_{67}, C_{61}),\ s_{69} = \gamma(s_{68}, C_{52})$
  $\}$ *of 22 predicates.*

The figure 4.23 provides a graphical representation of $BM(X_6(t))$ where the discernible states have replaced with the values X_{6i}, $i = 0...9$ of a three dimension vector X_6 corresponding to the values of x_5, ϕ_8 and x_6 (i.e. defined on $\Delta_5 \times \Delta_8 \times \Delta_6$). And again, the *symmetry* is complete with the values of x_5, the specificities of $X_6(t)$ being carried by the *asymmetry* introduced by X_6 and ϕ_8.

The model $M(X_6(t))$ of the dynamic process $X_6 = \{x_5, x_6, \phi_8\}$ is then the tuple $M(X_6(t)) =< PM(X_6(t)), SM(X_6(t)), FM(X_6(t)), BM(X_6(t)) >$.

Finally, it is to note that without the TOM4D notion of *network of dynamic process*, the global state space of this simple didactic example being directly linked with the number of values the variables can take, the size of the modeling state space is $2 \cdot 3 \cdot 3 \cdot 2 \cdot 2 \cdot 3 = 216$ (for

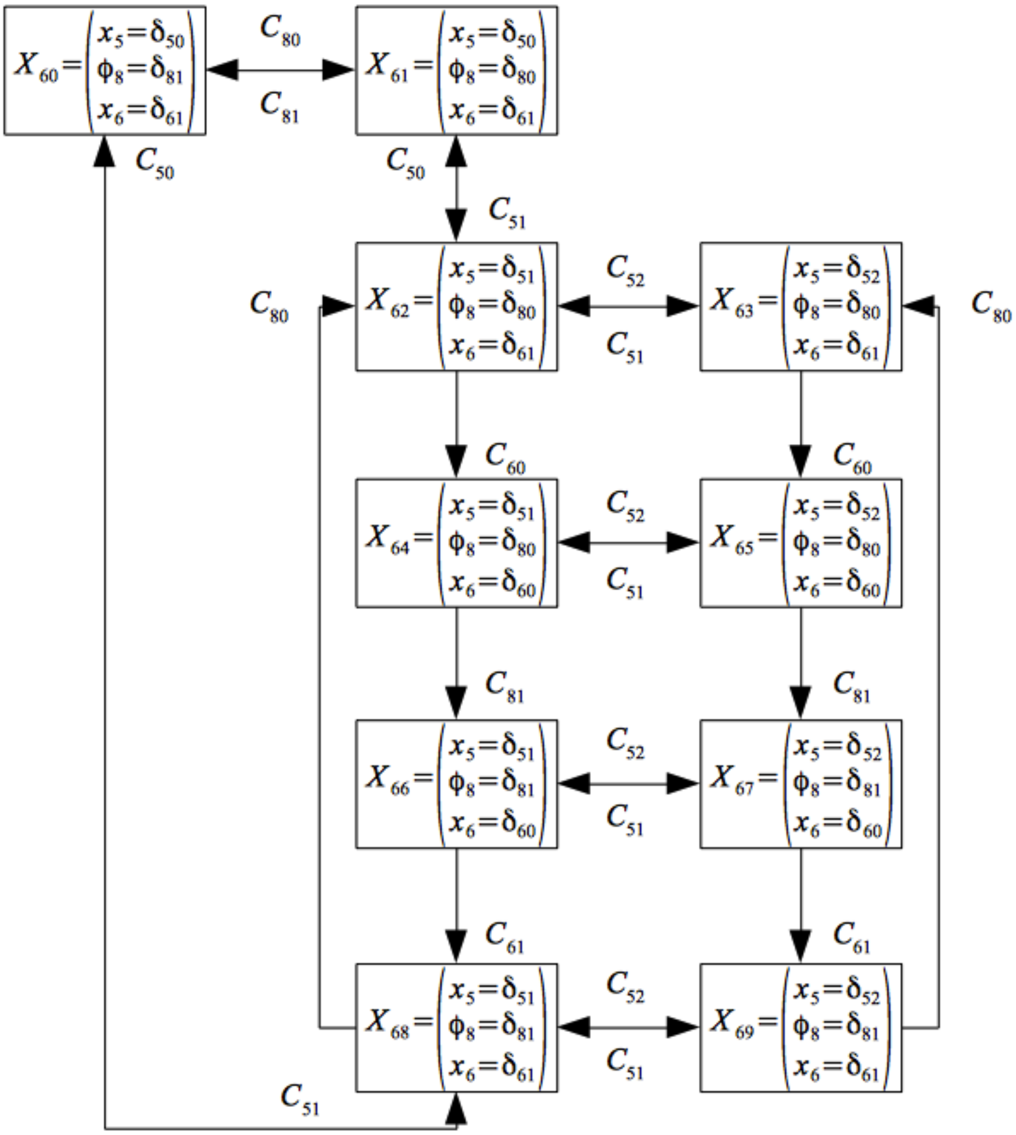

$X_{60}=\begin{pmatrix}x_5=\delta_{50}\\ \phi_8=\delta_{81}\\ x_6=\delta_{61}\end{pmatrix}$
$X_{61}=\begin{pmatrix}x_5=\delta_{50}\\ \phi_8=\delta_{80}\\ x_6=\delta_{61}\end{pmatrix}$
$X_{62}=\begin{pmatrix}x_5=\delta_{51}\\ \phi_8=\delta_{80}\\ x_6=\delta_{61}\end{pmatrix}$
$X_{63}=\begin{pmatrix}x_5=\delta_{52}\\ \phi_8=\delta_{80}\\ x_6=\delta_{61}\end{pmatrix}$
$X_{64}=\begin{pmatrix}x_5=\delta_{51}\\ \phi_8=\delta_{80}\\ x_6=\delta_{60}\end{pmatrix}$
$X_{65}=\begin{pmatrix}x_5=\delta_{52}\\ \phi_8=\delta_{80}\\ x_6=\delta_{60}\end{pmatrix}$
$X_{66}=\begin{pmatrix}x_5=\delta_{51}\\ \phi_8=\delta_{81}\\ x_6=\delta_{60}\end{pmatrix}$
$X_{67}=\begin{pmatrix}x_5=\delta_{52}\\ \phi_8=\delta_{81}\\ x_6=\delta_{60}\end{pmatrix}$
$X_{68}=\begin{pmatrix}x_5=\delta_{51}\\ \phi_8=\delta_{81}\\ x_6=\delta_{61}\end{pmatrix}$
$X_{69}=\begin{pmatrix}x_5=\delta_{52}\\ \phi_8=\delta_{81}\\ x_6=\delta_{61}\end{pmatrix}$
C_{80}
C_{81}
C_{50}
C_{50}
C_{51}
C_{80}
C_{52}
C_{51}
C_{80}
C_{60}
C_{60}
C_{52}
C_{51}
C_{81}
C_{81}
C_{52}
C_{51}
C_{61}
C_{61}
C_{52}
C_{51}
C_{51}

Figure 4.23: Generic Behavioral Model $BM(X_6(t))$

respectively x_1, x_2, x_5, ϕ_8, x_6 and x_7). And the potential number of state transition to examine is $216 \cdot 215 = 46\,440$!

This number illustrates the difficulty when modeling a dynamic process and consequently, one of the advantages of the TOM4D notion of *network of dynamic process*: the resulting model of the didactic example is made with only $8+9 = 17$ discernible states. Obviously, this simplicity of the model entails a little more complex diagnosis algorithms that will be described in the next chapter.

But, before examining these algorithms, it is important to examine one of the main advantages of the TOM4D methodology: the validation of the models.

4.5.4 Final Step: Model Validation

One of the aims of the TOM4D methodology is to provide models that can be validated by both the experts and the available data (cf. [PL10, PLG14] for an example of model validation from a database). With this didactic example, no expert and no datum is available. But the two scenarii ω_1 and ω_2 can be used to illustrates the way a database can be used to validate a TOM4D model.

The models $M(X_8)$, $M(X_6)$ and $M(X_7)$ of $X(t) = X_8(t) \cup X_7(t) \cup X_6(t)$ having be made from these two scenarii, it could be argue that they are *logical consequences* of these scenarii. This argument can be rejected only with the simple comparison of the models $M(\Omega)$ and $M(X(t))$: they differ strongly in all dimensions, that is to say the number and the role of the variables, the structure, the functions and the behavior of the process. Naturally, there exists some links between the two models but it is clear that the scenario model $M(\Omega)$ is essentially an important step before entering in the final modeling aiming at clarifying the experts propositions about the process, notably the propositions about its dynamics.

So, let us go back to the two scenarii:

1. $\omega_1 = \{C_{22}(t_1), C_{51}(t_2), C_{52}(t_3), C_{71}(t_4), C_{72}(t_5), C_{10}(t_6), C_{60}(t_7)\}$.

2. $\omega_2 = \{C_{22}(t_{10}), C_{51}(t_{11}), C_{52}(t_{12}), C_{71}(t_{13}), C_{72}(t_{14}), C_{21}(t_{15}), C_{51}(t_{16}), C_{20}(t_{17}), C_{50}(t_{18})\}$.

The *Superposition Theorem* 3.3 of the TOT allows to decompose these scenarii according to the three connected processes of $X(t)$:

1. $X_8(t) = \{x_1(t), x_2(t), x_5(t), \phi_8(t)\}$: only the occurrences of the observation classes C_{1y}, C_{2y}, C_{5y} and C_{8y} are concerned with $X_8(t)$.

2. $X_7(t) = \{x_5(t), x_7(t)\}$: only the occurrences of the observation classes C_{5y} and C_{7y} are concerned.

3. $X_6(t) = \{x_5(t), x_6(t), \phi_8(t)\}$: only the occurrences of the observation classes C_{5y}, C_{6y} and C_{8y} are concerned.

This distribution of the classes allows to define the sub-sequences that concern each of this three processes.

4.5.4.1 Validation with ω_2

Let us decomposes $\omega_2 = \{$ $C_{22}(t_{10})$, $C_{51}(t_{11})$, $C_{52}(t_{12})$, $C_{71}(t_{13})$, $C_{72}(t_{14})$, $C_{21}(t_{15})$, $C_{51}(t_{16})$, $C_{20}(t_{17})$, $C_{50}(t_{18})\}$ according to $X_8(t)$, $X_7(t)$ and $X_6(t)$ so that $\omega_2 = \omega_{28} \cup \omega_{27} \cup \omega_{26}$:

1. $\omega_{28} = \{C_{22}(t_{10}), C_{51}(t_{11}), C_{52}(t_{12}), C_{21}(t_{15}), C_{51}(t_{16}), C_{20}(t_{17}), C_{50}(t_{18})\}$.

2. $\omega_{27} = \{C_{51}(t_{11}), C_{52}(t_{12}), C_{71}(t_{13}), C_{72}(t_{14}), C_{51}(t_{16}), C_{50}(t_{18})\}$.

3. $\omega_{26} = \{C_{51}(t_{11}), C_{52}(t_{12}), C_{51}(t_{16}), C_{50}(t_{18})\}$.

The process $X_8(t)$ has no behavioral model. Its functional model is recalled in the figure 4.24 and defines two functions and the tables 4.19 and 4.20.

So, according to $FM(X_8(t))$:

1. $C_{22}(t_{10})$: $x_2 = \delta_{22} \Rightarrow x_5 = \delta_{52}$ by f_1. $C_{22}(t_{10})$ *explains* $C_{52}(t_{12})$.

2. $C_{51}(t_{11})$: $x_2 = \delta_{21} \Rightarrow x_5 = \delta_{51}$ by f_1. But ω_{28} doesn't contain a timed observation $C_{21}(t_k)$ at time t_k so that: $t_9 \leq t_k \leq t_{10}$ ($X_8(t)$ being a dynamic process, f_1 is supposed to be *causal*). So, a timed observation $C_{21}(t_k)$ *could be* lacking ω_1, or $FM(X_8(t))$ is *erroneous.*

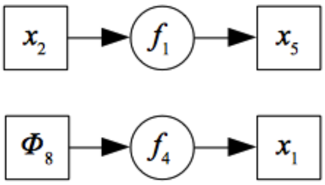


Figure 4.24: Generic Functional Model of $X_8(t)$

$f_1 : \Delta_{x_2} \to \Delta_{x_5}$		
x_2	$f_1(x_2)$	*Interpretation*
δ_{20}	δ_{50}	*the pump P is blocked* $\Rightarrow$ $flow_p$ *is zero*
δ_{21}	δ_{51}	*the pump P is leaking* $\Rightarrow$ $flow_p$ *is low*
δ_{22}	δ_{52}	*the pump P is ok* $\Rightarrow$ $flow_p$ *is normal*

Table 4.19: f_1 Generic Function of $FM(X_8(t))$

$f_4 : \Delta_{\phi_8} \to \Delta_{x_1}$		
ϕ_8	$f_4(\phi_8)$	*Interpretation*
δ_{80}	δ_{10}	$\phi_8 > 0$ *implies the pipe PI is leaking*
δ_{81}	δ_{11}	$\phi_8 = 0$ *implies the pipe PI is ok*

Table 4.20: f_4 Generic Function of $FM(X_8(t))$

3. $C_{21}(t_{15})$: $x_2 = \delta_{21} \Rightarrow x_5 = \delta_{51}$ by f_1. $C_{21}(t_{15})$ *explains* $C_{51}(t_{16})$.

4. $C_{20}(t_{17})$: $x_2 = \delta_{20} \Rightarrow x_5 = \delta_{50}$ by f_1. $C_{20}(t_{17})$ *explains* $C_{50}(t_{18})$.

So, three possibilities appears according to $FM(X_8(t))$ and ω_{28}:

1. $FM(X_8(t))$ is right and $C_{21}(t_k)$, $t_9 \leq t_k \leq t_{10}$, *would be* added in ω_2.

2. $FM(X_8(t))$ is erroneous and f_1 *would be* corrected.

3. $FM(X_8(t))$ is wrigh and the absence of the timed observation is explained by $M(X_7(t))$ or $M(X_6(t))$.

The validation must be continued to take a decision. The sequence ω_{17} concerns the process X_7 that has the behavioral model $BM(X_7(t))$ recalled in figure 4.25.

According $BM(X_7(t))$:

1. At time $t = t_9$, $X_7(t)$ is in the state $s(t_9) = s_{70}$ (i.e. $x_5 = \delta_{50}$ and $x_7 = \delta_{70}$).

2. $C_{51}(t_{11})$: $s(t_{11}) = s_{71}$, $x_5 = \delta_{51}$ and $x_7 = \delta_{70}$.

3. $C_{52}(t_{12})$: $s(t_{12}) = s_{73}$, $x_5 = \delta_{52}$ and $x_7 = \delta_{70}$.

4. $C_{71}(t_{13})$: $s(t_{13}) = s_{76}$, $x_5 = \delta_{52}$ and $x_7 = \delta_{71}$.

5. $C_{72}(t_{14})$: $s(t_{14}) = s_{78}$, $x_5 = \delta_{52}$ and $x_7 = \delta_{72}$.

6. $C_{51}(t_{16})$: $s(t_{16}) = s_{75}$, $x_5 = \delta_{51}$ and $x_7 = \delta_{72}$.

7. $C_{50}(t_{18})$: $s(t_{18}) = s_{77}$, $x_5 = \delta_{50}$ and $x_7 = \delta_{72}$.

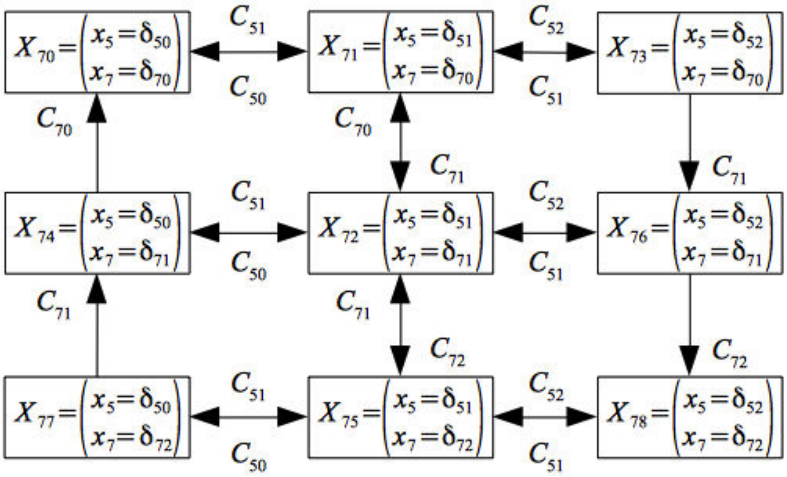

$X_{70}=\begin{pmatrix}x_5=\delta_{50}\\x_7=\delta_{70}\end{pmatrix}$
$X_{71}=\begin{pmatrix}x_5=\delta_{51}\\x_7=\delta_{70}\end{pmatrix}$
$X_{73}=\begin{pmatrix}x_5=\delta_{52}\\x_7=\delta_{70}\end{pmatrix}$
$X_{74}=\begin{pmatrix}x_5=\delta_{50}\\x_7=\delta_{71}\end{pmatrix}$
$X_{72}=\begin{pmatrix}x_5=\delta_{51}\\x_7=\delta_{71}\end{pmatrix}$
$X_{76}=\begin{pmatrix}x_5=\delta_{52}\\x_7=\delta_{71}\end{pmatrix}$
$X_{77}=\begin{pmatrix}x_5=\delta_{50}\\x_7=\delta_{72}\end{pmatrix}$
$X_{75}=\begin{pmatrix}x_5=\delta_{51}\\x_7=\delta_{72}\end{pmatrix}$
$X_{78}=\begin{pmatrix}x_5=\delta_{52}\\x_7=\delta_{72}\end{pmatrix}$
C_{51}
C_{50}
C_{52}
C_{51}
C_{70}
C_{70}
C_{71}
C_{71}
C_{51}
C_{50}
C_{52}
C_{51}
C_{71}
C_{71}
C_{72}
C_{72}
C_{51}
C_{50}
C_{52}
C_{51}

Figure 4.25: Generic Behavioral Model of $X_7(t)$

$BM(X_7(t))$ shows that the third possibility is validated if, and only if, $FM(X_8(t))$ and $BM(X_7(t))$ are wright: it is not necessary to add $C_{21}(t_k)$ in ω_2 until $FM(X_8(t))$ and $BM(X_7(t))$ are proved to be wrong.

The process $X_6(t)$ has also a behavioral model, recalled in figure 4.26.

According $M(X_8(t))$, $\omega_{26} = \{C_{51}(t_{11}), C_{52}(t_{12}), C_{51}(t_{16}), C_{50}(t_{18})\}$ leads to:

1. At time $t = t_9$, $X_6(t)$ is in the state $s(t_9) = s_{60}$ where $x_5 = \delta_{50}$, $\phi_8 = \delta_{81}$ and $x_6 = \delta_{61}$.

2. $C_{51}(t_{11})$: $s(t_{11}) = s_{68}$, $x_5 = \delta_{51}$, $\phi_8 = \delta_{81}$ and $x_6 = \delta_{61}$.

3. $C_{52}(t_{12}$: $s(t_{12}) = s_{69}$, $x_5 = \delta_{52}$, $\phi_8 = \delta_{81}$ and $x_6 = \delta_{61}$.

4. $C_{51}(t_{16})$: $s(t_{16}) = s_{68}$, $x_5 = \delta_{51}$, $\phi_8 = \delta_{81}$ and $x_6 = \delta_{61}$.

5. $C_{50}(t_{18})$: $s(t_{18}) = s_{60}$, $x_5 = \delta_{50}$, $\phi_8 = \delta_{81}$ and $x_6 = \delta_{61}$.

ω_2 is then *coherent* with $BM(X_8(t))$ and nothing can be said about the hypothesis of adding $C_{21}(t_k)$.

Let us examine briefly this hypothesis. The adding of this timed observation imposes that the set Δ_2 is *ordinal*: after $x_2 = \delta_2 0$ and before $x_2 = \delta_{22}$, the value δ_{21} *must be* assigned to x_2. But the original text said nothing to entail such a property for Δ_2 and this hypothesis has not been required to build $M(X(t))$.

So, unless to conclude that the models $M(X_8(t))$, $M(X_7(t))$ and $M(X_6(t))$ are together erroneous, ω_2 is *coherent* with $M(X(t))$.

4.5.4.2 Validation with ω_1

ω_1 is identical to ω_2 until the timed observation $C_{21}(t_{15})$ (cf. point 7. of ω_2).

ω_1 is a superposition of three sequences $\omega_{18} \cup \omega_{17} \cup \omega_{16}$:

1. $X_8(t) : \omega_{18} = \{C_{22}(t_1), C_{51}(t_2), C_{52}(t_3), C_{10}(t_6)\}$.

2. $X_7(t) : \omega_{17} = \{C_{51}(t_2), C_{52}(t_3), C_{71}(t_4), C_{72}(t_5)\}$.

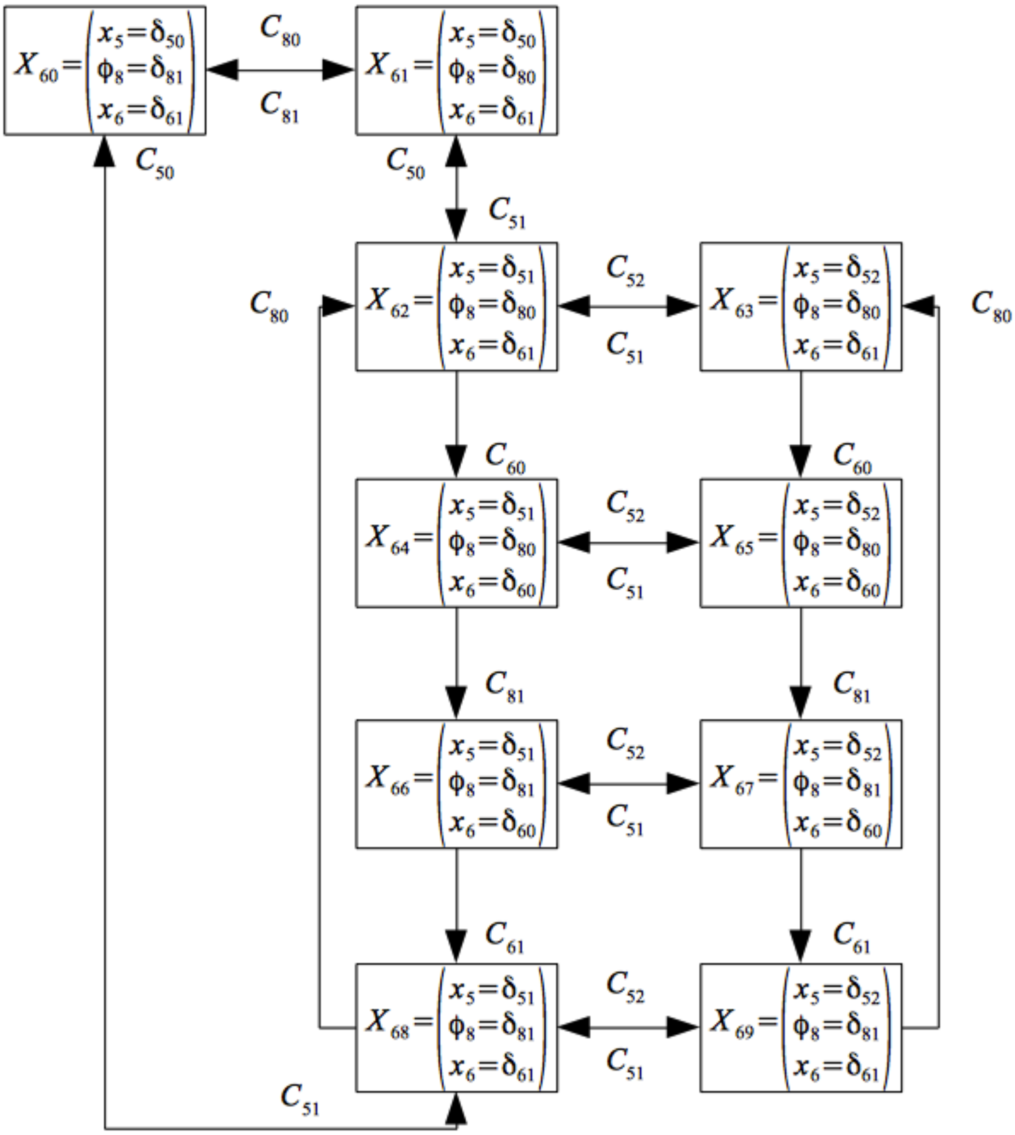

X60 = (x5=δ50, φ8=δ81, x6=δ61)
C80
C81
X61 = (x5=δ50, φ8=δ80, x6=δ61)
C50
C50
C51
C80
X62 = (x5=δ51, φ8=δ80, x6=δ61)
C52
C51
X63 = (x5=δ52, φ8=δ80, x6=δ61)
C80
C60
C60
X64 = (x5=δ51, φ8=δ80, x6=δ60)
C52
C51
X65 = (x5=δ52, φ8=δ80, x6=δ60)
C81
C81
X66 = (x5=δ51, φ8=δ81, x6=δ60)
C52
C51
X67 = (x5=δ52, φ8=δ81, x6=δ60)
C61
C61
X68 = (x5=δ51, φ8=δ81, x6=δ61)
C52
C51
X69 = (x5=δ52, φ8=δ81, x6=δ61)
C51

Figure 4.26: Generic Behavioral Model of $X_6(t)$

3. $X_6(t) : \omega_{16} = \{C_{51}(t_2), C_{52}(t_3), C_{60}(t_7)\}$.

The sequence ω_{18} has the following meaning according to $FM(X_8(t))$:

1. $C_{22}(t_1)$: $x_2 = \delta_{22} \Rightarrow x_5 = \delta_{52}$ by f_1. $C_{22}(t_1)$ *explains* $C_{52}(t_3)$.

2. $C_{51}(t_2)$: idem $C_{51}(t_{11})$ of ω_2.

3. $C_{10}(t_6)$: $\phi_8 = \delta_{80} \Rightarrow x_1 = \delta_{10}$ by f_4. As a consequence, f_4 and ω_{18} *subsumes* a timed observation $C_{80}(t_k)$ where $t_k \leq t_6$. In other words, $C_{80}(t_k)$ is an *hypothesis* that *could* explain $C_{10}(t_6)$ and then, *would be* added in ω_1.

This means:

- $FM(X_8(t))$ is erroneous and both f_1 and f_4 *would be* corrected.

- $FM(X_8(t))$ is right and $C_{21}(t_k)$, $t_0 \leq t_k \leq t_1$, *would be* added in ω_2 or its absence is explained by $M(X_7(t))$ or $M(X_6(t))$ (cf. ω_2).

- $FM(X_8(t))$ is right and a timed observation $C_{80}(t_k)$, $t_k \leq t_6$, *would explain* ω_{18}.

Let us consider the coherence between ω_{17} and $BM(X_7(t))$ (cf. figure 4.25):

1. At time $t = t_0$, $X_7(t)$ is in the state $s(t_0) = s_{70}$ (i.e. $x_5 = \delta_{50}$ and $x_7 = \delta_{70}$).

2. $C_{51}(t_2)$: $s(t_2) = s_{71}$, $x_5 = \delta_{51}$ and $x_7 = \delta_{70}$.

3. $C_{52}(t_3)$: $s(t_3) = s_{73}$, $x_5 = \delta_{52}$ and $x_7 = \delta_{70}$.

4. $C_{71}(t_4)$: $s(t_4) = s_{71}$, $x_5 = \delta_{52}$ and $x_7 = \delta_{71}$.

5. $C_{72}(t_5)$: $s(t_5) = s_{78}$, $x_5 = \delta_{52}$ and $x_7 = \delta_{72}$.

Obviously, as with ω_2 and for the same reason, the adding of a timed observation $C_{21}(t_k)$, $t_0 \leq t_k \leq t_1$, is not required.

The sequence ω_{16} concerns the process X_6 (cf. figure 4.23):

1. At time $t = t_0$, $s(t_0) = s_{60}$, $x_5 = \delta_{50}$, $\phi_8 = \delta_{81}$ and $x_6 = \delta_{61}$.

2. $C_{51}(t_2)$: $s(t_2) = s_{68}$, $x_5 = \delta_{51}$, $\phi_8 = \delta_{81}$ and $x_6 = \delta_{61}$.

3. $C_{52}(t_3)$: $s(t_3) = s_{69}$, $x_5 = \delta_{52}$, $\phi_8 = \delta_{81}$ and $x_6 = \delta_{61}$.

4. $C_{60}(t_7)$: there is no transition linking s_{69} to a state where $x_6 = \delta_{60}$. Except to go back to s_{68}, the only way to go out of this state is to go in state s_{63} with a timed observation of the class C_{80}.

So, either $BM(X_6(t))$ is wrong, either it leads to the *hypothesis* of an occurrence $C_80(t_k)$, $t_3 \leq t_k \leq t_7$, so that $s(t_k) = s_{63}$ (i.e. $x_5 = \delta_{52}$, $\phi_8 = \delta_{80}$ and $x_6 = \delta_{61}$). In that case, $C_{60}(t_7)$ allows the transition the discernible state $s(t_7) = s_{65}$ where $x_5 = \delta_{52}$, $x_6 = \delta_{60}$ and $\phi_8 = \delta_{80}$.

This leads to the following two hypothesis:

1. $M(X(t))$ is wright and the timed observation $C_{80}(t_{t_k})$, $t_k \leq t_6 \leq t_7$ is the *explanation* of ω_1: ϕ_8 being a variable added during the generic modeling step, no timed observation of the classes C_{8y} could appear in ω_1.

2. $M(X(t))$ is wrong according to ω_1.

Semantically, the timed observation $C_{80}(t_k)$ (i.e. $\phi_8 = \delta_{80}$) means $\phi_8(t_k) > 0$ that is to say, at time $t_k \leq t_6$, the surface of the *pipe PI* is not sufficient to conduct the all its input flow. This entails the *leaking* of the *pipe* and then the presence of water in the tank CO. This explanation seeming satisfactory on a semantic point of view, it is reasonable to infer the validity of $M(X(t))$

In other words, the two scenarii ω_1 and ω_1 validate the TOM4D model $M(X(t))$ of the didactic example of [CPR00].

4.6 Conclusion

TOM4D is the Knowledge Engineering methodology that has been designed to model dynamic processes with the aim of describing the process at the same level of abstraction an expert uses to realize a knowledge intensive task as monitoring, diagnosis and prognosis. TOM4D defines a way of using the modeling principles proposed by the Timed Observation Theory of [LG06].

This chapter presents the general steps of the TOM4D modeling process (knowledge interpretation, process definition and generic modeling), the TOM4D multi-modeling approach (structural, functional, behavioral, perception and network models) and the interpretation frameworks (CommonKADS, Formal Logic and Tetrahedron of States). This presentation is illustrated with the application of TOM4D on a didactic example of a simple hydraulic process firstly described in [CPR00].

The main advantages of TOM4D are the followings:

- Abstraction level.
  The TOM4D model resides at the same abstraction level as the expert's reasoning. In this sens, the TOM4D models are a representation of the way an expert *perceive* a dynamic process.

- Compatibility with the available data.
  TOM4D being based on the Timed Observation Theory, the available data can be used to build the model (cf. the scenario model step) but also to evaluate the coherence of the resulting models as illustrated in the preceding section.

- Symbol Driven.
  It can be quite surprising that a Knowledge Engineering methodology claims to be *symbol driven*: the aim is to avoid the introduction of erroneous interpretation in the models. This is particularly important when modeling a dynamic process because precisely, the description of its dynamics is not natural and so very difficult for both the experts and the Knowledge Engineers. The didactic example of [CPR00] clearly illustrates this point: it is striking to notice the difference between the scenario and the generic models.

- Granularity.
  The definition of a dynamic process as a *network of dynamic process* avoid the necessity of describing complex behavior in a unique behavioral model, most oftently unfeasible as the didactic example shows.

In [PL10], L. Pomponio introduces and formalizes the concept of *Abstraction Level* in the TOM4D methodology. Coupled with the notion of *network of dynamic processes*, these concepts provide a great modeling power, allowing to define complex processes at multiple abstraction levels, each level being described with a network of processes according a specific granularity. But this merging will introduce a new modeling complexity and news modeling problems.

So, before merging the two concepts, it is better to define the algorithms able to uses the TOM4D models within a knowledge intensive tasks as the diagnosis tasks. This is the aim of the next chapter where the same didactic example will be used to illustrate the way the algorithms work. The precedent section introduces the design of these algorithms: the *validation* of the

TOM4D model of the didactic example has been made through their *temporal simulation* with two scenarii. Such a simulation aims at predicting timed observations from those that are contained in the scenarii so that a comparison can be done. Such a comparison is the basis of any diagnosis algorithm.

Next, both TOM4D and the diagnosis algorithms will be used to diagnose a real world process, an hydraulic dam, in the application chapter.

CHAPTER 5

Diagnosis Process with Timed Observations

The previous chapter proposed a modeling methodology to diagnose dynamic processes, the Timed Observation Methodology for Diagnosis, based on the Timed Observation Theory, dedicated to the building of efficient model to diagnose dynamic processes from timed observations.

This chapter presents the algorithms that uses the different TOM4D models, the perception, the structural, the functional and the behavioral models of a network of dynamic processes, with the aim of diagnosing the undesirable states the processes of a network can reach over time. The specificity of the proposed algorithm is that it is based on an adequate transformation of the behavioral model $BM(X(t))$ of a network of dynamic processes $X(t) = \bigcup X_i(t)$ in a simple set of timed binary relations of the form $r_k(p, s, [\tau_{ps}^-, \tau_{ps}^+])$.

In other words, the proposed algorithm is based on the TOM4D concept of discernible state $s_i = \{..., r(p, s, [\tau_{ps}^-, \tau_{ps}^+]), ...\}$ defined as a simple set of timed binary relations $r(p, s, [\tau_{ps}^-, \tau_{ps}^+])$.

5.1 Diagnosing Dynamic Processes

The general idea of the diagnosis of a dynamic process $X(t)$ is to:

- *Detect* when the observed process reaches an undesirable state.
- *Explain* why this state is not desirable.
- *Describe* how the process reached this state.
- *Explain* why the process reached this state.

According to the Timed Observation Theory, this role is confided to an observation program $\Theta(X, \Delta)$ that observes continuously $X(t)$ (cf. the definition 3.1 of an observed process $(X(t), \Theta(X, \Delta))$).

The program $\Theta(X, \Delta)$ must have a cyclic working so that, at each cycle $k \in N$, it must :

1. Read the current value of each of the time functions $x_i(t_k)$ of $X(t)$ at $t = t_k$.
2. Apply the spatial segmentation principle to build the current $C(t_k)$ set of $n(t_k)$ timed observations $\{C_i(t_k)\}_{i=1...n(t_k)}$.
3. Define the current state $S(t_k)$ in which the process $X(t)$ is.
4. Detect if $S(t_k)$ is undesirable, and in that case:

(a) Explain why $S(t_k)$ is not desirable with a set of propositions $x_i(t_{k-n}) = \delta_i$ corresponding to an abnormal operation of R^{ab} defined in the perception model $PM(X(t))$ of $X(t)$.

(b) Describe how $X(t)$ reaches $S(t_k)$ with an adequate sequence $\omega(t_k)$ of timed observations.

(c) Explain why the process reached $S(t_k)$ with an abstract chronicle model $M(t_k) = \{r_k(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])\}_{k=1...n}$ of n temporal binary relations $r_k(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ (cf. definition 3.6) representing an adequate *model of faulty behavior*.

The distribution of the timed observations $C_i(t_k)$ of an input sequence $\omega(t_k)$ according to the structure of the *model of faulty behavior* $M(t_k)$ constitutes an *instantiated fault model*. In other words, the diagnosis tasks of an observation program $\Theta(X, \Delta)$ is mainly based on the real-time building of *instantiated fault model*. Generally speaking, a model of the dynamic process, even very weak, is required to execute these tasks of *Detection*, *Explanation* and *Description*. Within the TOT framework, the diagnosis will uses the TOM4D behavioral model $BM(X(t))$ (cf. definition 4.4) of the dynamic process to diagnose. As a consequence, the proposed diagnosis algorithm is fundamentally *Timed Observation Driven*.

This chapter is dedicated to the presentation of the algorithms that are required to implement the main tasks of such an observation program $\Theta(X, \Delta)$. To this aim, the two scenarii ω_1 and ω_2 will be used as a mean to illustrate the main mechanisms that are required by these algorithms. The execution of an observation program $\Theta(X, \Delta)$ will then be simulated with these scenarii and the TOM4D model $M(X(t))$ of the hydraulic process of [Con00] with the TOM4D model $M(X(t))$ of the preceding chapter. Naturally, because these scenarii are directly expressed in terms of timed observations, the steps 1 and 2 of the work cycle of $\Theta(X, \Delta)$ will not be illustrated.

So, before entering into the presentation, let us recall the main elements of the TOM4D model

5.2 Main Elements about $M(X(t))$

The main elements defined by $M(X(t))$ are recalled in the figures 5.1 for $SM(X(t))$, 5.2 for $FM(X(t))$, 5.3 for $BM(X(t))$, with the tables 5.1 (variables and observations classes) and 5.2 (functions f_1 and f_4).

Variable	Values	X	Definition Domain	Classes
$mode_{PI}$	$leaking, ok$	x_1	$\Delta_{x_1} = \{\delta_{10}, \delta_{11}\}$	$C_1 = \{C_{10}, C_{11}\}$
$mode_P$	$blocked, leaking, ok$	x_2	$\Delta_{x_2} = \{\delta_{20}, \delta_{21}, \delta_{22}\}$	$C_2 = \{C_{20}, C_{21}, C_{22}\}$
$flow_p$	zro_p, low_p, nrm_p	x_5	$\Delta_{x_5} = \{\delta_{50}, \delta_{51}, \delta_{52}\}$	$C_5 = \{C_{50}, C_{51}, C_{52}\}$
$level_{CO}$	pre_{CO}, abs_{CO}	x_6	$\Delta_{x_6} = \{\delta_{60}, \delta_{61}\}$	$C_6 = \{C_{60}, C_{61}\}$
$level_{TA}$	$zro_{TA}, low_{TA}, nrm_{TA}$	x_7	$\Delta_{x_7} = \{\delta_{70}, \delta_{71}, \delta_{72}\}$	$C_7 = \{C_{70}, C_{71}, C_{72}\}$
$Surface_{PI}^{-1}$	$\phi_8 > 0, \phi_8 = 0$	ϕ_8	$\Delta_{\phi_8} = \{\delta_{80}, \delta_{81}\}$	$C_8 = \{C_{80}, C_{81}\}$

Table 5.1: Variables and Observation Classes

The behavioral model $BM(X(t))$ of $X(t) = X_8(t) \cup X7(t) \cup X_6(t)$ is the union of the behavioral models of it sub-processes $BM(X(t)) = BM(X_8(t)) \cup BM(X_7(t)) \cup BM(X_6(t))$:

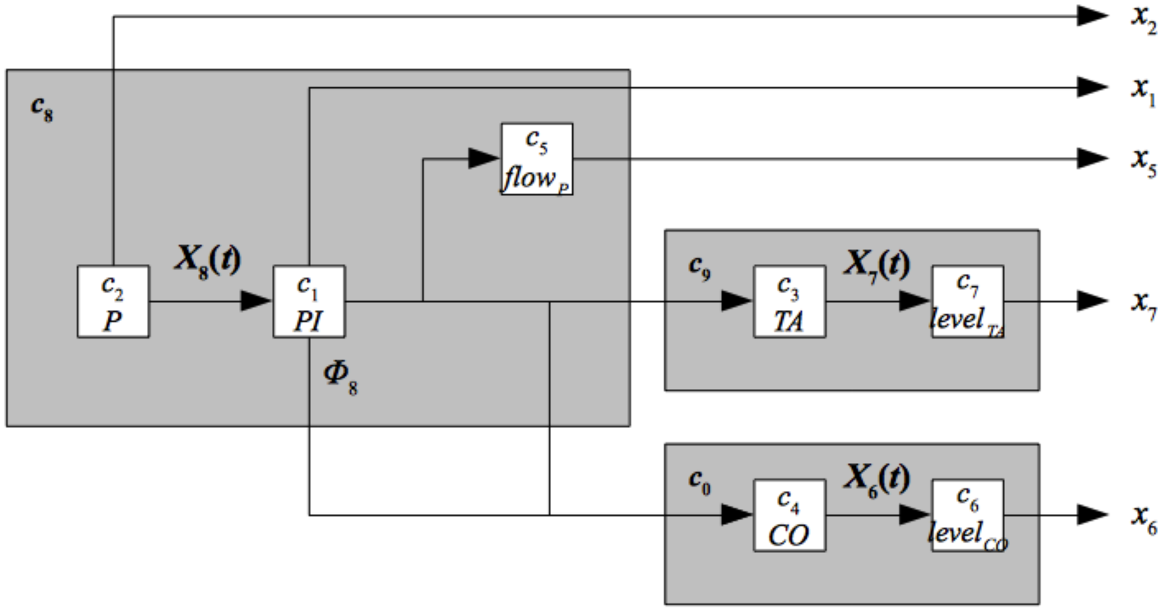


Figure 5.1: Generic Structural Model of $X(t)$

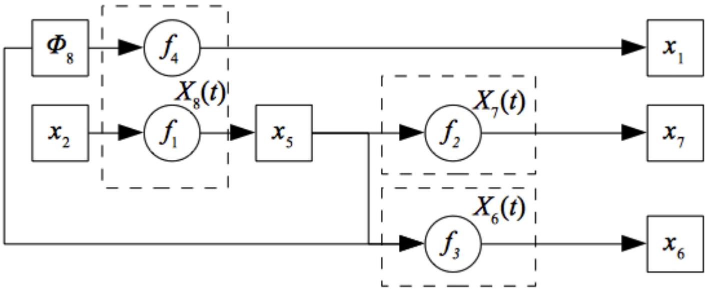


Figure 5.2: Generic Functional Model of $X(t)$

$f_1 : \Delta_{x_2} \rightarrow \Delta_{x_5}$		
x_2	$f_1(x_2)$	*Interpretation*
δ_{20}	δ_{50}	$\delta_{20} \Rightarrow \delta_{50}$
δ_{21}	δ_{51}	$\delta_{21} \Rightarrow \delta_{51}$
δ_{22}	δ_{52}	$\delta_{22} \Rightarrow \delta_{52}$

$f_4 : \Delta_{\phi_8} \rightarrow \Delta_{x_1}$		
ϕ_8	$f_4(\phi_8)$	*Interpretation*
δ_{80}	δ_{10}	$\phi_8 > 0 \Rightarrow \delta_{10}$
δ_{81}	δ_{11}	$\phi_8 = 0 \Rightarrow \delta_{11}$

Table 5.2: Definition of the Generic Functions f_1 and f_4 of $X_8(t)$

- $S = S_8 \cup S_7 \cup S_6 = \{$
  $s_{60} = \{r_{601}(C_{50}, C_{51}), r_{602}(C_{50}, C_{80}), r_{603}(C_{81}, C_{51}), r_{604}(C_{81}, C_{80})\}$,
  $s_{61} = \{r_{611}(C_{50}, C_{51}), r_{612}(C_{50}, C_{81}), r_{613}(C_{80}, C_{51}), r_{614}(C_{80}, C_{81})\}$,
  $s_{62} = \{r_{621}(C_{51}, C_{50}), r_{622}(C_{51}, C_{52}), r_{623}(C_{51}, C_{60}), r_{624}(C_{80}, C_{50}), r_{625}(C_{80}, C_{52}), r_{626}(C_{80}, C_{60})\}$,
  $s_{63} = \{r_{631}(C_{52}, C_{51}), r_{632}(C_{52}, C_{60}), r_{633}(C_{80}, C_{51}), r_{634}(C_{80}, C_{60})\}$,
  $s_{64} = \{r_{641}(C_{60}, C_{52}), r_{642}(C_{60}, C_{81}), r_{643}(C_{51}, C_{52}), r_{644}(C_{51}, C_{81})\}$,
  $s_{65} = \{r_{651}(C_{52}, C_{51}), r_{652}(C_{52}, C_{81}), r_{653}(C_{60}, C_{51}), r_{654}(C_{60}, C_{81})\}$,
  $s_{66} = \{r_{661}(C_{51}, C_{52}), r_{662}(C_{51}, C_{61}), r_{663}(C_{81}, C_{52}), r_{664}(C_{81}, C_{61})\}$,
  $s_{67} = \{r_{671}(C_{52}, C_{51}), r_{672}(C_{52}, C_{61}), r_{673}(C_{81}, C_{51}), r_{674}(C_{81}, C_{61})\}$,
  $s_{68} = \{r_{681}(C_{51}, C_{50}), r_{682}(C_{51}, C_{52}), r_{683}(C_{51}, C_{80}), r_{684}(C_{61}, C_{50}), r_{685}(C_{61}, C_{52}), r_{686}(C_{61}, C_{80})\}$,

$s_{69} = \{r_{691}(C_{52}, C_{51}), r_{692}(C_{52}, C_{80}), r_{693}(C_{61}, C_{51}), r_{694}(C_{61}, C_{80})\}$
$s_{70} = \{r_{701}(C_{50}, C_{51}), r_{702}(C_{70}, C_{51})\}$,
$s_{71} = \{r_{711}(C_{51}, C_{50}), r_{712}(C_{51}, C_{52}), r_{713}(C_{51}, C_{71}), r_{714}(C_{70}, C_{50}), r_{715}(C_{70}, C_{52}), r_{716}(C_{70}, C_{71})\}$,
$s_{72} = \{r_{721}(C_{51}, C_{50}), r_{722}(C_{51}, C_{52}), r_{723}(C_{51}, C_{70}), r_{724}(C_{51}, C_{72}),$
$r_{725}(C_{71}, C_{50}), r_{726}(C_{71}, C_{52}), r_{727}(C_{71}, C_{70}), r_{728}(C_{71}, C_{72})\}$,
$s_{73} = \{r_{731}(C_{52}, C_{71}), r_{732}(C_{52}, C_{51})\}$,
$s_{74} = \{r_{741}(C_{50}, C_{51}), r_{742}(C_{50}, C_{70}), r_{743}(C_{71}, C_{51}), r_{744}(C_{71}, C_{70})\}$,
$s_{75} = \{r_{751}(C_{51}, C_{50)}, r_{752}(C_{51}, C_{52}), r_{753}(C_{51}, C_{71}), r_{754}(C_{72}, C_{50}), r_{755}(C_{72}, C_{52}), r_{756}(C_{72}, C_{71})\}$,
$s_{76} = \{r_{761}(C_{52}, C_{51}), r_{762}(C_{52}, C_{72}), r_{763}(C_{71}, C_{51}), r_{764}(C_{71}, C_{72})\}$,
$s_{77} = \{r_{771}(C_{50}, C_{51}), r_{772}(C_{50}, C_{71})\}$,
$s_{78} = \{r_{781}(C_{52}, C_{51}), r_{782}(C_{72}, C_{51})\}$,
$s_{80} = \{r_{801}(C_{20}, C_{50})\}$,
$s_{81} = \{r_{811}(C_{21}, C_{51})\}$,
$s_{82} = \{r_{820}(C_{22}, C_{52})\}$,
$s_{83} = \{r_{831}(C_{80}, C_{10})\}$,
$s_{84} = \{r_{841}(C_{81}, C_{11})\}$,
$\}$ of 24 discernible states defining 85 timed binary relations.

- The set $C = C_8 \cup C_7 \cup C_6 = \{$
  $C_{10} = \{(x_{10}, \delta_{10})\}, C_{11} = \{(x_{10}, \delta_{11})\}$,
  $C_{20} = \{(x_2, \delta_{20})\}, C_{21} = \{(x_2, \delta_{21})\}, C_{22} = \{(x_2, \delta_{22})\}$,
  $C_{50} = \{(x_5, \delta_{50})\}, C_{51} = \{(x_5, \delta_{51})\}, C_{52} = \{(x_5, \delta_{52})\}$,
  $C_{60} = \{(x_6, \delta_{60})\}, C_{61} = \{(x_6, \delta_{61})\}, C_{62} = \{(x_6, \delta_{62})\}$,
  $C_{70} = \{(x_7, \delta_{70})\}, C_{71} = \{(x_7, \delta_{71})\}, C_{72} = \{(x_7, \delta_{72})\}$,
  $C_{80} = \{(\phi_8, \delta_{80})\}, C_{81} = \{(\phi_8, \delta_{81})\}$
  $\}$ made of 16 observations classes.

- The set $R^s = R^{s_7} \cup R^{s_6} = \{$
  $s_{60} = \gamma(s_{61}, C_{81}),\ s_{60} = \gamma(s_{68}, C_{50}),$
  $s_{61} = \gamma(s_{60}, C_{80}),\ s_{61} = \gamma(s_{62}, C_{50}),$
  $s_{62} = \gamma(s_{61}, C_{51}),\ s_{62} = \gamma(s_{63}, C_{51}),\ s_{62} = \gamma(s_{68}, C_{80})\},$
  $s_{63} = \gamma(s_{62}, C_{52}),\ s_{63} = \gamma(s_{69}, C_{80}),$
  $s_{64} = \gamma(s_{62}, C_{60}),\ s_{64} = \gamma(s_{65}, C_{51}),$
  $s_{65} = \gamma(s_{63}, C_{60}),\ s_{65} = \gamma(s_{64}, C_{52}),$
  $s_{66} = \gamma(s_{64}, C_{81}),\ s_{66} = \gamma(s_{67}, C_{51}),$
  $s_{67} = \gamma(s_{65}, C_{81}),\ s_{67} = \gamma(s_{66}, C_{52}),$
  $s_{68} = \gamma(s_{66}, C_{61}),\ s_{68} = \gamma(s_{69}, C_{51}),\ s_{68} = \gamma(s_{60}, C_{51})\},$
  $s_{69} = \gamma(s_{67}, C_{61}),\ s_{69} = \gamma(s_{68}, C_{52})$
  $s_{70} = \gamma(s_{71}, C_{50}),\ s_{70} = \gamma(s_{74}, C_{70}),$
  $s_{71} = \gamma(s_{70}, C_{51}),\ s_{71} = \gamma(s_{73}, C_{51}),\ s_{71} = \gamma(s_{72}, C_{70}),$
  $s_{73} = \gamma(s_{71}, C_{52}),$
  $s_{74} = \gamma(s_{72}, C_{50}),\ s_{74} = \gamma(s_{77}, C_{71}),$
  $s_{72} = \gamma(s_{71}, C_{71}),\ s_{72} = \gamma(s_{76}, C_{51}),\ s_{72} = \gamma(s_{75}, C_{71}),\ s_{72} = \gamma(s_{74}, C_{51}),$
  $s_{76} = \gamma(s_{72}, C_{52}),\ s_{76} = \gamma(s_{73}, C_{71}),$
  $s_{77} = \gamma(s_{75}, C_{50}),$

$s_{75} = \gamma(s_{72}, C_{72})$, $s_{75} = \gamma(s_{78}, C_{51})$, $s_{75} = \gamma(s_{77}, C_{51})$,
$s_{78} = \gamma(s_{76}, C_{72})$, $s_{78} = \gamma(s_{75}, C_{52})$,
$\}$ of 41 predicates.

The figure 5.3 provides a graphical representation of $BM(X(t))$.

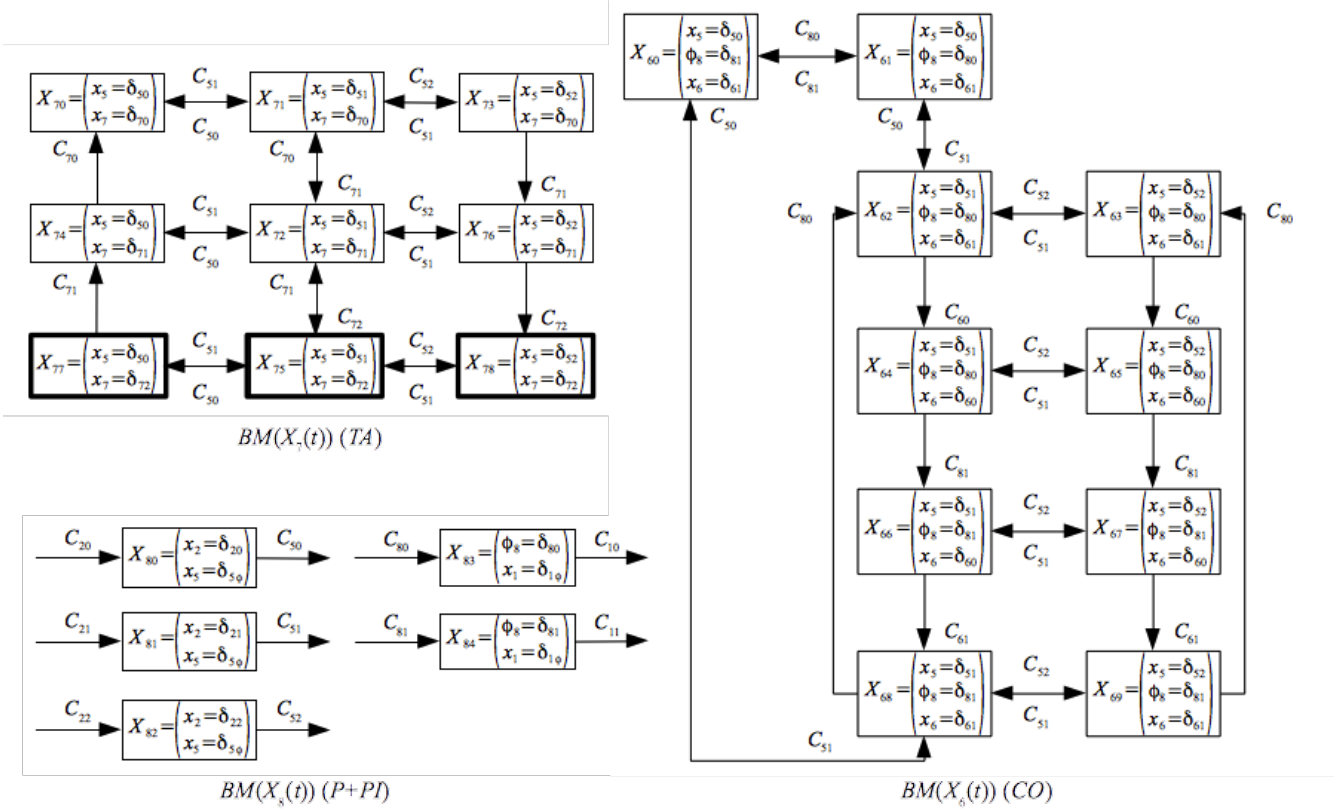

$X_{70} = (x_5 = \delta_{50}, x_7 = \delta_{70})$
$X_{71} = (x_5 = \delta_{51}, x_7 = \delta_{70})$
$X_{73} = (x_5 = \delta_{52}, x_7 = \delta_{70})$
$X_{74} = (x_5 = \delta_{50}, x_7 = \delta_{71})$
$X_{72} = (x_5 = \delta_{51}, x_7 = \delta_{71})$
$X_{76} = (x_5 = \delta_{52}, x_7 = \delta_{71})$
$X_{77} = (x_5 = \delta_{50}, x_7 = \delta_{72})$
$X_{75} = (x_5 = \delta_{51}, x_7 = \delta_{72})$
$X_{78} = (x_5 = \delta_{52}, x_7 = \delta_{72})$
C_{51}
C_{50}
C_{52}
C_{70}
C_{71}
C_{72}
$BM(X_7(t))$ (TA)
C_{20}
$X_{80} = (x_2 = \delta_{20}, x_5 = \delta_{59})$
C_{50}
C_{21}
$X_{81} = (x_2 = \delta_{21}, x_5 = \delta_{59})$
C_{51}
C_{22}
$X_{82} = (x_2 = \delta_{22}, x_5 = \delta_{59})$
C_{52}
C_{80}
$X_{83} = (\phi_8 = \delta_{80}, x_1 = \delta_{10})$
C_{10}
C_{81}
$X_{84} = (\phi_8 = \delta_{81}, x_1 = \delta_{10})$
C_{11}
$BM(X_8(t))$ (P+PI)
$X_{60} = (x_5 = \delta_{50}, \phi_8 = \delta_{81}, x_6 = \delta_{61})$
$X_{61} = (x_5 = \delta_{50}, \phi_8 = \delta_{80}, x_6 = \delta_{61})$
$X_{62} = (x_5 = \delta_{51}, \phi_8 = \delta_{80}, x_6 = \delta_{61})$
$X_{63} = (x_5 = \delta_{52}, \phi_8 = \delta_{80}, x_6 = \delta_{61})$
$X_{64} = (x_5 = \delta_{51}, \phi_8 = \delta_{80}, x_6 = \delta_{60})$
$X_{65} = (x_5 = \delta_{52}, \phi_8 = \delta_{80}, x_6 = \delta_{60})$
$X_{66} = (x_5 = \delta_{51}, \phi_8 = \delta_{81}, x_6 = \delta_{60})$
$X_{67} = (x_5 = \delta_{52}, \phi_8 = \delta_{81}, x_6 = \delta_{60})$
$X_{68} = (x_5 = \delta_{51}, \phi_8 = \delta_{81}, x_6 = \delta_{61})$
$X_{69} = (x_5 = \delta_{52}, \phi_8 = \delta_{81}, x_6 = \delta_{61})$
C_{80}
C_{81}
C_{50}
C_{51}
C_{52}
C_{60}
C_{61}
$BM(X_6(t))$ (CO)

Figure 5.3: Generic Behavioral Model of $X(t)$

The perception models $PM(X_8(t))$ and $PM(X_6(t))$ (cf. definitions 4.9 and 4.17) providing no operating goals for $X_8(t)$ and $X_6(t)$, the goal of the operations of $X(t)$ concerns uniquely its sub-process $X_7(t)$ (cf. $PM(X_7(t))$ of definition 4.13):

- $R^{goal_7} = \{\exists t_i, t_i \geq t_0, \forall t \geq t_i, x_7(t) \geq \Psi_{72}\}$

The normal operations R^n of $X(t)$ are distributed over $X_8(t)$, $X_7(t)$ and $X_6(t)$:

- $R^{n_8} = \{\forall t \geq t_0, x_1(t) = \delta_{11} \wedge x_2(t) = \delta_{22}\}$.
- $R^{n_7} = \{\forall t \geq t_i, x_7(t) \geq \Psi_{72}\}$
- $R^{n_6} = \{\forall t \geq t_0, x_6(t) \leq \Psi_{60}\}$

Similarly, the abnormal operations R^{ab} of $X(t)$ are the followings:

- $R^{ab_8} = \{\forall t \geq t_0, x_1(t) = \delta_{10} \vee x_2(t) = \delta_{20} \vee x_2(t) = \delta_{21}\}$.
- $R^{ab_7} = \{\forall t \geq t_i, x_7(t) < \Psi_{72}\}$
- $R^{ab_6} = \{\forall t \geq t_0, x_6(t) > \Psi_{60}\}$

5.3 Identification of the Possible Current States

The global cycle of the observation program $\Theta(X, \Delta)$ is based on the identification of the current state $S(t_k)$ in which the process $X(t)$ is (step 3 of the observation program $\Theta(X, \Delta)$)).

Let us suppose that at a time t_k of the cycle of $\Theta(X, \Delta)$, the steps 1 and 2 have be done so that a set $\{..., C_{ij}(t_k), ...\}$ of timed observations $C_{ij}(t_k)$ has been written in a database, represented by a sequence ω of timed observation in the TOT framework. This means that at the time cycle t_k, the program $\Theta(X, \Delta)$ *write* the set $\{..., C_{ij}(t_k), ...\}$ of timed observations and *add* it in the sequence ω.

The followings *computing* functions can then be defined:

Definition 5.1 *Write Computing Function*
The function write$(\omega, \{..., C_{ij}(t_k), ...\})$ updates a sequence ω with a set $\{..., C_{ij}(t_k), ...\}$ of timed observations.

- *write*$(\omega, \{..., C_{ij}(t_k), ...\}) \equiv \omega(t_k) = \omega(t_{k-1}) \cup \{..., C_{ij}(t_k), ...\}$.

Definition 5.2 *Read Computing Function*
The function read(ω, t_k) provides the set $\{..., C_{ij}(t_k), ...\}$ of timed observations added in ω at the time cycle t_k:

- *read*$(\omega, t_k) = \{..., C_{ij}(t_k), ...\}$.

To simplify, when there is no confusion, the following abuses of language can be made:

$$write(\omega, \{..., C_{ij}(t_k), ...\}) \equiv write(\omega, ..., C_{ij}(t_k), ...) \equiv \omega(t_k) = \omega(t_{k-1}) + ... + C_{ij}(t_k) + ... \quad (5.1)$$

$$read(\omega, t_k) = \{..., C_{ij}(t_k), ...\} \equiv ..., C_{ij}(t_k), ... \quad (5.2)$$

According to the definition 3.2 of a timed observation, the meaning of $C_{ij}(t_k)$ is an assignation $x_i(t_k) = \delta_{ij}$ (cf. equation 3.8) and, according to the *Discernible State* modelling principle of the TOT:

- *Such an assignation results necessarily of an observable modification in the dynamic process $X(t)$. So two occurrences $C_i(t_k)$ and $C_j(t_{k+1})$ marks an observable state transition in an observed process $(X(t), \Theta(X, \Delta))$. This means that a temporal binary relation $r(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ defines a particular discernible state.*

The problem of finding the current state of $X(t)$ is then to find, in the set S of the behavioral model $BM(X(t))$ of $X(t)$, the set $S(t_k)$ of the discernible states s_k in which $X(t)$ can be at t_k that *explains* the timed observations $C_{ij}(t_k)$ returned by *read*(ω, t_k). Yet, the set $S(t_k)$ is logically *constrained* by the equal predicates $s_j = \gamma(s_i, C_k)$ contained in R^s of the behavioral model $BM(X(t))$: only the states allowing the corresponding *write*$(\omega, \{..., C_{ij}(t_{k-1}), ...\})$ must be considered.

Indeed, according to the TOM4D methodology, a discernible state s_n is simply a set of timed binary relations $r_{nm}(C_{k_i}, C_{k_j}), [\tau_{k_{ij}}^-, \tau_{k_{ij}}^+])$. As a consequence, given a timed observation

$C_k(t_k)$ of an observation class C_k, finding the set $S(t_k)$ of the corresponding discernible states s_m is finding, in the set $S = \bigcup_i S_i$ of the discernible states defined by $BM(X(t))$, all the states s_m containing a particular binary relation of the form $r_{nm}(C_{ij}, C_{jk}, [\tau_n^- \tau_n^+])$ where C_{ij} or C_{jk} corresponds to C_k. So, when denoting C_ϕ the class variable representing any class $C_{xy} \in C$, given a timed observation $C_k(t_k)$, two sets of states must be build, the set $S^s(k)$ of *successors* states and the set $S^p(k)$ of *predecessors* states:

1. $\forall s \in S, \exists r_{sm}(C_{ij}, C_\phi, [\tau_n^- \tau_n^+]) \in S, C_{ij} = C_k \Rightarrow s \in S^s(k)$.

2. $\forall s \in S, \exists r_{sm}(C_\phi, C_{ij}, [\tau_n^- \tau_n^+]) \in S, C_{ij} = C_k \Rightarrow s \in S^p(k)$.

For example, the timed observation $C_{51}(t_k)$ allows to build the following sets of successors and predecessor states at time t_k (cf. figure 5.3 or definitions 4.16 and 4.20):

- $\forall s \in S_7, r_x(C_{51}, C_\phi) \in S_7 \Rightarrow s \in S_7^s(k)$: $S_7^s(k) = \{\ s_{71},\ s_{72},\ s_{75}\ \}$.

- $\forall s \in S_7, r_x(C_\phi, C_{51}) \in S_7 \Rightarrow s \in S_7^p(k)$: $S_7^p(k) = \{\ s_{70},\ s_{73},\ s_{74},\ s_{76},\ s_{77},\ s_{78}\ \}$.

- $\forall s \in S_6, r_x(C_{51}, C_\phi) \in S_6 \Rightarrow s \in S_6^s(k)$: $S_6^s(k) = \{\ s_{60},\ s_{62},\ s_{64},\ s_{66},\ s_{68}\ \}$.

- $\forall s \in S_6, r_x(C_\phi, C_{51}) \in S_6 \Rightarrow s \in S_6^p(k)$: $S_6^p(k) = \{\ s_{60},\ s_{61},\ s_{63},\ s_{65},\ s_{67},\ s_{69}\ \}$.

Let us suppose that $read(\omega, t_{t_k-1}) = C_{71}(t_{k-1})$ at the preceding time cycle t_{k-1}. The timed observation $C_{71}(t_{k-1})$ allows to build the set $S_7^s(k-1)$ containing the possible *successors* states at time t_{k-1} for $X_7(t)$, $X_6(t)$ being not concerned with $C_{71}(t_{k-1})$:

- $\forall s \in S_7, r_x(C_{71}, C_\phi) \in S_7 \Rightarrow s \in S_7^s(k-1)$: $S_7^s(k-1) = \{\ s_{72},\ s_{74},\ s_{76}\ \}$.

In other words, knowing $C_{51}(t_k)$ allows to *filter* $S_7^p(k)$: at t_k the only possible predecessor states at t_{k-1} are contained in the set $S_7^p(k) \cap S_7^s(k-1)$, that is to say $\{\ s_{74},\ s_{76}\ \}$. And knowing the state s_{72} has been removed from $S_7^p(k)$ allows also to *remove* from $S_7^s(k)$ any state s_x for which $BM(X_7(t))$ contains a predicate $s_x = \gamma(s_{72}, C_{51}(t_k))$. In the example, it is easy to see that the states 71 and 74 will be removed from $S_7^s(k)$ so that after the removing, $S_7^s(k)$ contains only one state s_{72}.

This reasoning is quite complex: using the fact that a state is simple set of timed binary relations simplifies the reasoning and makes it much more efficient.

5.4 Observation Model

The central idea of the *Timed Observation Driven algorithm for Diagnosis* (TOD4D algorithm) is to replace the discernible states with their corresponding set of timed binary relations and to directly uses them to diagnose.

The key point is to note that the set R^s of a behavioral model $BM(X(t))$ provides a *structure* to the set S of discernible states, that is to say to the corresponding set $R = \{r_{nm}(C_{ij}, C_{jk}, [\tau_{nm}^- \tau_{nm}^+])\}$ of timed binary relations:

- $\forall s_k \in S, \forall r_{ki} \in s_k, r_{ki} \in R$

Similarly to the γ modeling function defined on $C \times S \times S$, a γ_r modeling function can be defined on $C \times R \times R$, that is to say at the level of the timed binary relations. Such a function aims at representing the relation between the timed binary relations of a behavioral model $BM(X(t))$ in order to build an *Observation Model*:

Definition 5.3 *Observation Model*
An observation model $OM(X(t))$ of a dynamic process $X(t)$ is an abstract chronicle model made with a set $R = \{r_m(C_i^m, C_j^m, [\tau_{ij}^-, \tau_{ij}^+])\}$ of n_R timed binary relations $r_m(C_i^m, C_j^m, [\tau_{ij}^-, \tau_{ij}^+])$ between two observation classes C_i^m and C_j^m of C^l, $i \neq j$, where the classes C_i^m and C_j^m are labelled with the index of the timed binary relation.

To transform a behavioral model $BM(X(t)) =< S, C, R^s >$ into its corresponding observation model $OM(X(t))$, the following three rules must be implemented, the rules 1 and 2 concerning a uniquely state and the rule 3 concerning two different states:

1. Rule 1:
   $\forall r_1(C_{1i}, C_{1j}, [\tau_{1ij}^-, \tau_{1ij}^+]) \in s_n, \exists r_2(C_{2i}, C_{2j}, [\tau_{2ij}^-, \tau_{2ij}^+]) \in s_n \wedge C_{1i} = C_{2i} \Rightarrow$
   $\exists r_1(C_{1i}^n, C_{1j}^n, [\tau_{1ij}^-, \tau_{1ij}^+]) \in R, \exists r_2(C_{2i}^n, C_{2j}^n, [\tau_{2ij}^-, \tau_{2ij}^+]) \in R, C_{1i}^n = C_{2i}^n$

2. Rule 2:
   $\forall r_1(C_{1i}, C_{1j}, [\tau_{1ij}^-, \tau_{1ij}^+]) \in s_n, \exists r_2(C_{2i}, C_{2j}, [\tau_{2ij}^-, \tau_{2ij}^+]) \in s_n \wedge C_{1j} = C_{2J} \Rightarrow$
   $\exists r_1(C_{1i}^n, C_{1j}^n, [\tau_{1ij}^-, \tau_{1ij}^+]) \in R, \exists r_2(C_{2i}^n, C_{2j}^n, [\tau_{2ij}^-, \tau_{2ij}^+]) \in R, C_{1j}^n = C_{2j}^n$

3. Rule 3:
   $\forall s_m = \gamma(s_n, C_i) \in R^s, \exists r_1(C_{1i}^n, C_{1j}^n, [\tau_{1ij}^-, \tau_{1ij}^+]) \in R \wedge \exists r_2(C_{2i}^m, C_{2j}^m, [\tau_{2ij}^-, \tau_{2ij}^+]) \in R, \Rightarrow$
   $C_{1j}^n = C_{2i}^m$

The three rules corresponds to the three types of transformations of figure 5.4 within an abstract chronicle model.

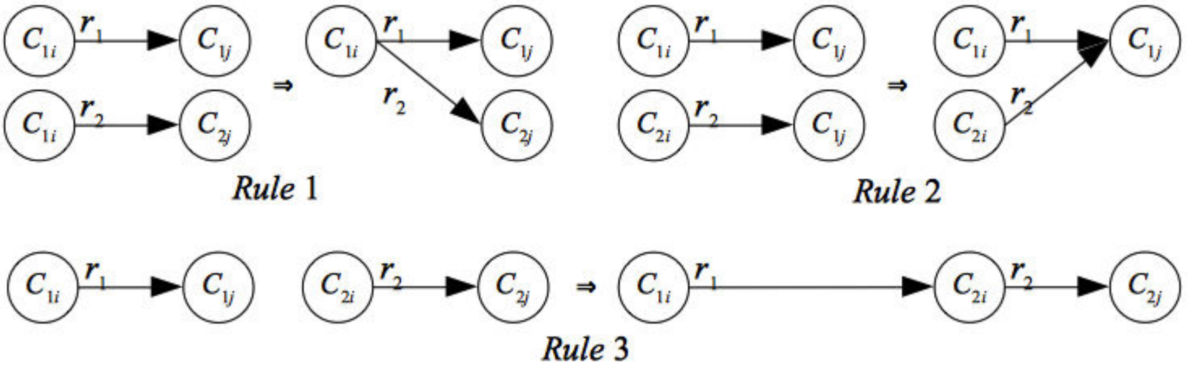


Figure 5.4: Three rules

The following algorithms aims at building an observation model $OM(X(t))$ from a behavioral model $BM(X(t)) =< S, C, R^s >$. In these algorithms, a timed binary relation is of the generic form $r(p, s, [\tau^-, \tau^+])$, where p and s denote respectively the *predecessor* observation class C_i and the *successor* class C_j.

Algorithm: setClassLabels
Inputs:
A behavioral model $BM(X(t)) =< S, C, R^s >$.
A set $s = \{r_i(C_1, C_2, [\tau_{12}^-, \tau_{12}^+])\}$ of timed binary relations.
Output:
An observation model $OM(X(t)$ made with a set $R = \{r_i(C_1^i, C_2^i, [\tau_{12}^-, \tau_{12}^+])$
of labelled timed binary relations
Method:
For all states s of S, set a label to the classes of each timed binary relation r_i,
and set r_i in R.
The classes are labelled with the index i of the corresponding timed binary relation r_i.

```
1. $R = \{\}$
2. $\forall i = 1...S.size()$ //For all the states
2.1 $s = S(i)$
2.2 $\forall j = 1...s.size()$ //For all the relations
2.2.1 $r = s(i)$
2.2.2 $p = r.getP()$ //Get the preceding observation class
2.2.3 $p.setLabel(r.getLabel())$
2.2.4 $s = r.getS()$
2.2.5 $s.setLabel(r.getLabel())$
2.2.6 $r.setP(p)$
2.2.7 $r.setS(s)$
2.2.8 $R = R + r$
3. return $R$
```

Such an observation model $OM(X(t)$ can be graphically represented with a graph $G(X(t)) = (C_g, R_g)$ where:

- $C_g = \{C_i^n\}_{i=1...n_c}$ is a set of n_{C_g} nodes corresponding to the observations classes of an observation model $OM(X(t))$.
- $R_g = \{r_k(C_i^n, C_j^m, [\tau_{ij}^-, \tau_{ij}^+])\}$ is a set of n_{R_g} links corresponding to the timed binary relations contained in the set R of $OM(X(t))$.

The figures 5.5 and 5.6 present the graph $G(X(t))$ corresponding to the observation model $OM(X(t))$ made with the *toObservationModel* algorithm applied on the behavioral model $BM(X(t))$ of the hydraulic process. In particular, it can be seen that $OM(X(t))$ is also divided in two sub-models: $OM(X(t)) = OM(X_{78}(t)) \cup OM(X_{68}(t))$ so that $OM(X_{78}(t)) \cap OM(X_{68}(t)) = \Phi$.

The suite of this chapter will show that using such an observation model drastically simplifies the diagnosis of dynamic process. But to this aim, the step 4 of the observation program $\Theta(X, \Delta)$ requires to detect if the current state $S(t_k)$ of $X(t)$ is satisfactory or not.

5.5 Operating Goals and Unsatisfactory States

To define if the current state $S(t_k)$ of a process $X(t)$ is unsatisfactory, it is necessary to *compare* $S(t_k)$ with the propositions of R^{ab}.

The propositions contained in R^{ab} being expressed in terms of equality predicates of the form $x_1(t) = \delta_{10}$ or $x_7(t) < \Psi_{72}$ for examples, it is necessary beforehand to build a corresponding set

Algorithm: shareWithinStates
Inputs:
A set $S = \{r_i(C_1, C_2, [\tau_{12}^-, \tau_{12}^+])\}$ of a behavioral model $BM(X(t)) =< S, C, R^s >$.
A set $R = \{r_i(C_1^i, C_2^i, [\tau_{12}^-, \tau_{12}^+])$ of an observation model $OM(X(t))$
Output:
The partially pruned set R
Method:
For all states s of S, apply the rule 1 and 2 on R.

```
1. $\forall i = 1...S.size()$ //For all the states
1.1 $s = S(i)$
1.2 $\forall j = 1...s.size()$ //For all the relations
1.2.1 $r1 = s(i)$
1.2.2 $\forall j = j...s.size()$ //For all the other relations

1.2.2.1 $r2 = s(j)$
//Rule 1
1.2.2.1 $r1.getP() = r2.getP() \Rightarrow$
1.2.2.1.1 $labelledR1 = R.get(r1)$ //Find in R the labelled relation corresponding to r1
1.2.2.1.2 $labelledR2 = R.get(r2)$ //Find in R the labelled relation corresponding to r2
1.2.2.1.3 $p = labelledR1.getP()$
1.2.2.1.4 $labelledR2.setP(p)$
//Rule 2
1.2.2.2 $r1.getS() = r2.getS() \Rightarrow$
1.2.2.2.1 $labelledR1 = R.get(r1)$
1.2.2.2.2 $labelledR2 = R.get(r2)$
1.2.2.2.3 $s = labelledR1.getS()$
1.2.2.2.4 $labelledR2.setS(s)$
3. return $R$
```

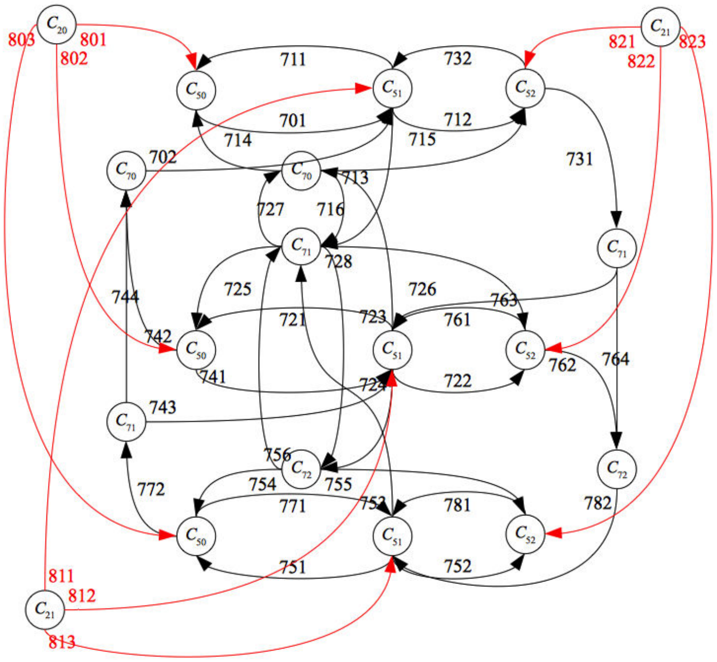


Figure 5.5: Graph of $OM(X_{78}(t))$

Algorithm: shareBeetweenStates
Inputs:
The set $R^s = \{r_2 = \gamma(r_1, C_i)\}$ of a behavioral model $BM(X(t)) =< S, C, R^s >$.
A set $R = \{r_i(C_1^i, C_2^i, [\tau_{12}^-, \tau_{12}^+])$ of an observation model $OM(X(t))$
Output:
The connected set R.
Method:
For all predicate equal of R^s, apply the rule 3 on R.
A state is a set of

```
1. $\forall i = 1...R^s.size()$ //For all the equal predicate
1.1 $e = R^s(i)$
1.2 $s1 = e.getS1()$ //State 1
1.2 $s2 = e.getS2()$ //State 2
1.3 $c = e.getC()$ //Observation class
//Find the labelled class c2i of the labelled s2 corresponding to c //c2i is unique
1.4 $found = false$
1.5 $\forall j = 1...s2.size()$ //For all the relations of State 2
1.5.1 $r2 = s2(j)$
1.5.2 IF $found = false$ and $r2.getP() = c$ THEN
1.5.2.1 $labelledR2 = R.getP(r2)$ //Find in R the labelled relation corresponding to r2
1.5.2.1 $c2i = labelledR2.getP()$
1.5.2.1 $found = true$
//Replace the labelled class c1j of the labelled s1 with with c2i
1.6 $\forall j = 1...s1.size()$ //For all the relations of State 1
1.6.1 $r = s1(j)$
1.6.2 $cj = r1.getS()$
1.6.3 IF $cj = c$ THEN
1.6.3.1 $labelledR1 = R.get(r1)$ //Find in R the labelled relation corresponding to r1
1.6.3.2 $labelledR1.setS(c2i)$
2. return $R$
```

Algorithm: toObservationModel
Input:
A behavioral model $BM(X(t)) =< S, C, R^s >$.
Output:
The set R of an observation model $OM(X(t))$

```
1. $R = setClassLabels(BM(X(t)))$
2. $R = shareWithinStates(S, R)$
3. $R = shareBeetweenStates(R^s, R)$
4. return $R$
```

S^{ab} containing the undesirable states of $X(t)$. A state s_k being a set of timed binary relations $\{r(C_{k_i}, C_{k_j}), [\tau_{k_{ij}}^-, \tau_{k_{ij}}^+]\}$, some rules must be defined to deduce the undesirable states of $X(t)$ from the abnormal operations R^{ab} of $X(t)$. These rules are based on the interpretation of an observation class $C_i = \{(x_i, \delta_{ij})\}$ under the form: $C_i \equiv x_i(t_k) = \delta_{ij}$. The propositions of R^{ab} containing forms like $x_1(t) = \delta_{10}$ can then be directly formulated in terms of observation class, C_{10} in the example. But a relation must be done between the threshold (Ψ_{72}) for propositions containing forms like $x_7(t) < \Psi_{72}$. It can be noticed that this link is *implicit* within the TOM4D

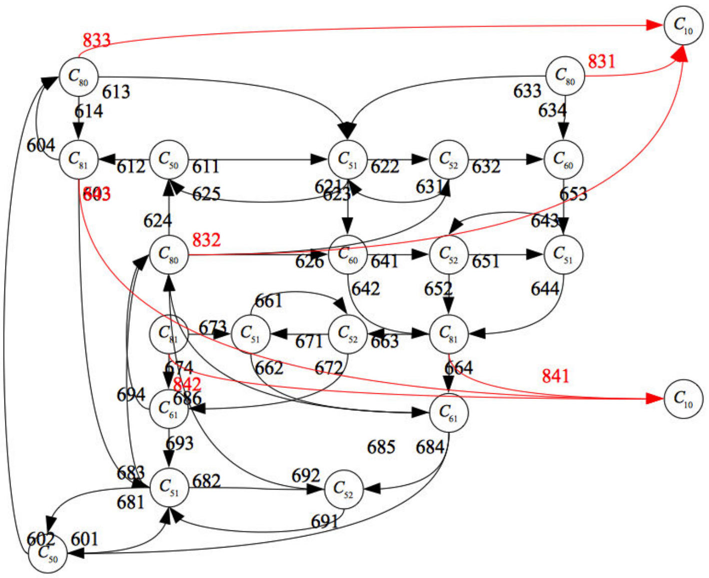


Figure 5.6: Graph of $OM(X_{68}(t))$

formalism: simply, the relations between the thresholds and the ranges they defined can be provided as a comment in the models (remember that a constant like δ_{72} denotes the range of values $[\Psi_{72}, +\infty[$ for $x_7(t)$). This interpretation allows to define the followings rules for $X_8(t)$ and $X_6(t)$ where $S = S_8 \cup S_6$:

1. $\forall s \in S, \exists r(C_\phi, C_{10}, [\phi, \phi]) \in s \Rightarrow s \in S^{ab}\ (X_8(t))$.
2. $\forall s \in S, \exists r(C_{20}, C_\phi, [\phi, \phi]) \in s \Rightarrow s \in S^{ab}\ (X_8(t))$.
3. $\forall s \in S, \exists r(C_{21}, C_\phi, [\phi, \phi]) \in s \Rightarrow s \in S^{ab}\ (X_8(t))$.
4. $\forall s \in S, \exists r(C_{60}, C_\phi, [\phi, \phi]) \in s \Rightarrow s \in S^{ab}\ (X_6(t))$.

For example, it is easy to see that the undesirable states for $X_8(t)$ are s_{83}, s_{80} and s_{81} (cf. figure 5.3), and s_{64} and s_{65} for $X_6(t)$.

The problem differs for $X_7(t)$: the timed constraint linked with t_i in the proposition $\forall t \geq t_i, x_7(t) < \Psi_{72}$ of R^{ab_7} imposes to *determine* the time t_i *before* identifying the undesirable states of $X_7(t)$. The time t_i is defined by the proposition $\exists t_i, t_i \geq t_0, \forall t \geq t_i, x_7(t) \geq \Psi_{72}$ of R^{goal_7}, that is to say by the general goal of the operation of $X(t)$: a state containing a binary relation of the form $r(C_{ij}, C_\phi, [\phi, \phi])$ or $r(C_\phi, C_{ij}, [\phi, \phi])$ can be or cannot be an undesirable state according to the time at which it is evaluated. In other words, reaching a state s at a time t_k can be a good or bad news according to the *past* of the dynamic process. For example, reaching the state s_{72} at time t_4 is a good news after the state sequence: $s(t_0) = s_{70}$, $s(t_1) = s_{71}$, $s(t_2) = s_{73}$ and $s(t_3) = s_{76}$. But reaching the same state s_{72} at time t_6 after the sequence $s(t_0) = s_{70}$, $s(t_1) = s_{71}$, $s(t_2) = s_{73}$, $s(t_3) = s_{76}$, $s(t_4) = s_{72}$ and $s(t_5) = s_{75}$ is clearly a bad news because the state $s_7 6$ is one of the state defined by R^{goal_7} (the two others being s_{77} and s_{78}, cf. the bold boxes of $BM(X_7(t)$ in figure 5.3)). This property is a fundamental characteristic of dynamic processes: generally speaking, the undesirables states of a dynamic process depends of its recent behavior, that is to say its current *trajectory* in its state space.

This justifies a specific mechanism dedicated to the management of the time t_i that allows to determine dynamically if a state is undesirable or not. In the example, such a mechanism aims at managing the current set $S^{ab}(t_k)$ according to the followings rules:

Rule 1: $\forall t_k, \forall s \in S, t_k = t_0 \wedge \exists r(C_\phi, C_{10}, [\phi, \phi]) \in s \Rightarrow s \in S^{ab}(t_k)$ $(X_8(t))$.

Rule 2: $\forall t_k, \forall s \in S, t_k = t_0 \wedge \exists r(C_{20}, C_\phi, [\phi, \phi]) \in s \Rightarrow s \in S^{ab}(t_k)$ $(X_8(t))$.

Rule 3: $\forall t_k, \forall s \in S, t_k = t_0 \wedge \exists r(C_{21}, C_\phi, [\phi, \phi]) \in s \Rightarrow s \in S^{ab}(t_k)$ $(X_8(t))$.

Rule 4: $\forall t_k, \forall s \in S, t_k = t_0 \wedge \exists r(C_\phi, C_{60}, [\phi, \phi]) \in s \Rightarrow s \in S^{ab}(t_k)$ $(X_6(t))$.

Rule 5: $\forall t_k, \forall s \in S, \exists t_i, t_i \geq t_0 \wedge t_k > t_i \wedge \exists r(C_\phi, C_{70}, [\phi, \phi]) \in s \Rightarrow s \in S^{ab}(t_k)$ $(X_7(t))$.

Rule 6: $\forall t_k, \forall s \in S, \exists t_i, t_i \geq t_0 \wedge t_k > t_i \wedge \exists r(C_\phi, C_{71}, [\phi, \phi]) \in s \Rightarrow s \in S^{ab}(t_k)$ $(X_7(t))$.

These rules imposes the right management of the values of the times t_0 and t_i. In particular, when t_0 is defined once, t_i can take a lot of values. For example, according to the state sequence $s(t_0) = s_{70}$, $s(t_1) = s_{71}$, $s(t_2) = s_{73}$, $s(t_3) = s_{76}$, $s(t_4) = s_{72}$, $s(t_5) = s_{75}$, $s_{72}(t_6)$, $s_{76}(t_7)$, $s_{78}(t_7)$, $s_{75}(t_8)$ and $s_{77}(t_9)$, t_i is undefined before t_5, takes a first value at t_5, is undefined again at t_6 and takes a second value at t_7. t_i is then defined from t_7 to the end of the sequence.

The use of the observation model $OM(X(t)) = OM(X_{78}(t)) \cup OM(X_{68}(t))$ simplifies the detection of the abnormal states with the *a priori* identification of the corresponding relations:

- Rule 1 $(OM(X_{68}(t)))$: r_{831}, r_{832}, r_{833}.
- Rule 2 $(OM(X_{78}(t)))$: r_{801}, r_{802}, r_{803}.
- Rule 3 $(OM(X_{78}(t)))$: r_{811}, r_{812}, r_{813}.
- Rule 4 $(OM(X_{68}(t)))$: r_{623}, r_{626}, r_{632}, r_{634}.
- Rule 5 $(OM(X_{78}(t)))$: r_{723}, r_{727}, r_{744}.
- Rule 6 $(OM(X_{78}(t)))$: r_{715}, r_{716}, r_{721}, r_{731}, r_{756}, r_{772}.

Distributed over $OM(X_{78}(t))$ and $OM(X_{68}(t))$, these relations allows to define two sets of relations corresponding to abnormal states:

- $OM(X_{68}(t))$ (Rules 1 and 4): $R^{ab}_{68} = \{\ r_{831}, r_{832}, r_{833}, r_{623}, r_{626}, r_{632}, r_{634}\ \}$.
- $OM(X_{78}(t))$ (Rules 2, 3, 5 and 6): $R^{ab}_{78} = \{\ r_{801}, r_{802}, r_{803}, r_{811}, r_{812}, r_{813}, r_{723}, r_{727}, r_{744}, r_{715}, r_{716}, r_{721}, r_{731}, r_{756}, r_{772}\ \}$.

In other words, any path containing one of these relation *can mean* that the process $X(t)$ reached an abnormal state: the relations linked with the rules 5 and 6 depends of t_i. This leads to define two sets of relations corresponding to abnormal states:

- Rules 1, 2, 3 and 4:
  $R^{ab}_{static} = \{\ r_{831}, r_{832}, r_{833}, r_{801}, r_{802}, r_{803}, r_{811}, r_{812}, r_{813}, r_{623}, r_{626}, r_{632}, r_{634}\ \}$.

- Rules 5 and 6:
  $R^{ab}_{dynamic} = \{\ r_{723},\ r_{727},\ r_{744},\ r_{715},\ r_{716},\ r_{721},\ r_{731},\ r_{756},\ r_{772}\ \}$.

This leads to the adequate management of the set $R^{ab}_{current}$ (t_ϕ denotes an unknown value):

- $\forall t_k, t_k \geq t_0 \Rightarrow R^{ab}_{current} = R^{ab}_{static}$.
- $\forall t_k, t_i \neq t_\phi \wedge t_k \geq t_i \Rightarrow R^{ab}_{current} = R^{ab}_{static} \cup R^{ab}_{dynamic}$.

The detection of an abnormal state can the be made with a computing function $isSatisfactory(r_k)$, $r_k \in OM(X(t))$, returning $false$ if and only if $r_i \in R^{ab}_{current}$.

Function: isSatisfactory
Inputs:
A binary relation r_k.
The set $R^{ab}_{current}$ of binary relations corresponding to abnormal states.
Output:
False, iff $r_k \in R^{ab}_{current}$.

```
1. $\forall i = 1...R^{ab}_{current}.size()$
1.1. $r = R^{ab}_{current}(i)$
1.2. IF $r_k = r$ THEN return false
2. return true
```

A computing function $getRelations(C_p, C_s)$ is also required in order to get the set $s = \{..., r_{ps}, ...\}$ of relations r_{ps} corresponding to the pair of observation classes C_p and C_s in an observation model $OM(X(t))$.

Function: $getRelations(C_p, C_s, R)$
Inputs:
A pair of observation classes p and s.
The set R of an observation model $OM(X(t))$.
Output:
A set S of binary relation r_{ps} of the form $r_k(p, s, [\tau^-_{ps}, \tau^+_{ps}])$ $s \subseteq R$.

```
1. $S = \{\}$
2. $\forall i = 1...R.size()$
2.1. $r = R(i)$
2.2. $p = r.getP()$
2.3. $s = r.getS()$
2.4. IF $C_p = p$ AND $C_s = s$ THEN $S = S + r$
3. return S
```

Finally, a computing function $isAPath(p, r)$ is necessary in order to build the set $P = \{..., p_i, ...$ of pathes $p_i = \{..., r^ik, ...\}_{k=1...n_i}$ of n_{p_i} binary relations r^ik corresponding to a sequence of timed observation ω. This function simply check if the last relation $r^i_{n_i}$ of a path p_i share the same labelled observation class with the relation r and then are connected in an observation model $OM(X(t))$.

Function: isAPath
Inputs:
A relation r and a path *path*.
Output:
True iff the class s of the last relation of *path* is equal to the class p of r.
This function supposes that a *path* is implemented in an adequate structure so that the tail of a path is given by the *tail* function.

```
1. $tail = path.tail()$
2. IF $tail.getS() = r.getP()$ return true
2. return false
```

5.6 Diagnosis of ω_1 and ω_2

So, the general mechanisms to evaluate a state being defined, its possible to analyze the way of doing a diagnosis with the sequences ω_1 and ω_2 of timed observations. To the aim of defining an algorithm as general as possible, let us assume the two followings hypothesis about these sequences:

- The initial state of $X(t)$ is not known, and
- At the end of each of the sequences, the process is supposed to be in an unsatisfactory state that justifies the request to a diagnosis.

5.6.1 Sequences ω_1 and ω_2 of Timed Observations

The figure 5.7 shows that the hydraulic system contains three sensors, graphically represented through *eyes* symbols:

- A sensor measuring the variable $flow_p$, denoted x_5 in the TOM4D model, associated with a set of three observations classes $C_5 = \{(x_5, \delta_{50})\}$, $C_{51} = \{(x_5, \delta_{51})\}$, $C_{52} = \{(x_5, \delta_{52})\}$ respectively linked with $flow_p$'s values zro_p, low_p and nrm_p.
- A sensor measuring the variable $level_{CO}$, denoted x_6 and associated with a set of two observations classes $C_6 = \{(x_6, \delta_{60})\}$, $C_{61} = \{(x_6, \delta_{61})\}$ respectively linked with $level_{CO}$'s values pre_{CO} and abs_{CO}.
- A sensor measuring the variable $level_{TA}$, denoted x_7 and associated with a set of three observations classes $C_7 = \{(x_7, \delta_{70})\}$, $C_{71} = \{(x_7, \delta_{71})\}$, $C_{72} = \{(x_7, \delta_{72})\}$ respectively linked with $level_{TA}$'s values zro_{TA}, low_{TA} and nrm_{TA}.

Because only the outputs of the sensors x_5, x_6 and x_7 can be recorded, only the occurrences of observation classes made with the values of the variables x_5, x_6 and x_7. The scenarii ω_1 and ω_2 becomes then two sequences of timed observations:

- $\omega_1 = \{C_{51}(t_2), C_{52}(t_3), C_{71}(t_4), C_{72}(t_5), C_{60}(t_7)\}$:
    1. t_1: no timed observation ($x_2(t_1) = \delta_{22}$: the pump is turn on).
    2. t_2: $C_{51}(t_2)$ ($x_5(t_2) = \delta_{51}$, the output flow of the pump is low).

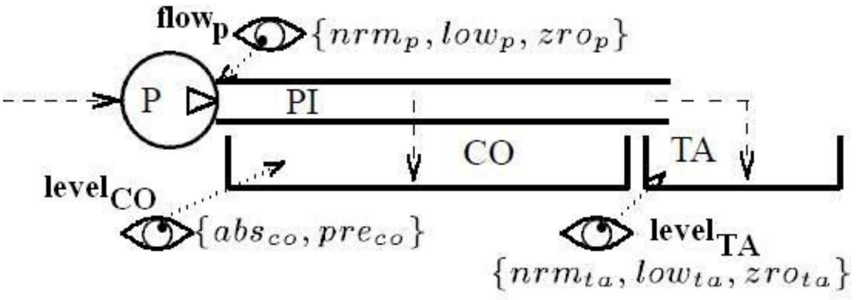


Figure 5.7: Graphical Representation of the Hydraulic System of [CPR00]

3. t_3: $C_{52}(t_3)$ ($x_5(t_3) = \delta_{52}$, the output flow of the pump reaches the normal value).
4. t_4: $C_{71}(t_3)$ ($x_7(t_4) = \delta_{71}$, the level in the tank TA achieves the low level).
5. t_5: $C_{72}(t_5)$ ($x_7(t_5) = \delta_{72}$, the level in the tank TA achieves the normal level).
6. t_6: no timed observation ($x_1(t_6) = \delta_{10}$, the pipe PI is leaking).
7. t_7: $C_{60}(t_7)$ ($x_6(t_7) = \delta_{60}$, the tank CO contains water).

- $\omega_2 = \{C_{51}(t_{11}), C_{52}(t_{12}), C_{71}(t_{13}), C_{72}(t_{14}), C_{51}(t_{16}), C_{50}(t_{18})\}$:

1. t_{10}: no timed observation ($x_2(t_{10}) = \delta_{22}$, the pump is turn on).
2. t_{11}: $C_{51}(t_{11})$ ($x_5(t_{11}) = \delta_{51}$, the output flow of the pump is low).
3. t_{12}: $C_{52}(t_{12})$ ($x_5(t_{12}) = \delta_{52}$, the output flow of the pump reaches the normal value).
4. t_{13}: $C_{71}(t_{13})$ ($x_7(t_{13}) = \delta_{71}$, the level in the tank TA achieves the low level).
5. t_{14}: $C_{72}(t_{14})$ ($x_7(t_{14}) = \delta_{72}$, the level in the tank TA achieves the normal level).
6. t_{15}: no timed observation ($x_2(t_{15}) = \delta_{21}$, the pump P is leaking).
7. t_{16}: $C_{51}(t_{16})$ ($x_5(t_{16}) = \delta_{51}$, the output flow of the pump enters again in the low range).
8. t_{17}: no timed observation ($x_2(t_{17}) = \delta_{20}$, the pump P is blocked).
9. t_{18}: $C_{50}(t_{18})$ ($x_5(t_{18}) = \delta_{50}$, the flow in the pipe enters again in the zero range).

The first important idea of the diagnosis algorithm is to *explain* the faults with occurrences $C_i(t_k)$ of observation classes C_i that are not linked with sensors. In other words, it is to find the *unobserved* occurrences that *could explain* the current state of the process and to place them in a set of paths (cf. definition 3.7) describing the way the process reached an unsatisfactory state. The second important idea is to use the observation model $OM(X(t))$ (cf. figures 5.5 and 5.6) in place of the behavioral model $BM(X(t))$ (cf. figure 5.3) to build the paths. These paths will be used in order to build a sequences of state vector values $\{..., X_i(t_k), ...\}$ that can be used to identify, if required, the corresponding states.

To this aim, let us define the state vector $X(t)$ of equation 5.3.

$$X(t) = \left\{ \begin{array}{rcl} x_1(t) & = & \delta_{\phi_1} \\ x_2(t) & = & \delta_{\phi_2} \\ x_5(t) & = & \delta_{\phi_5} \\ x_6(t) & = & \delta_{\phi_6} \\ x_7(t) & = & \delta_{\phi_7} \\ \phi_8(t) & = & \delta_{\phi_8} \end{array} \right\} \tag{5.3}$$

Each time t_k an occurrence $C_i(t_k)$ is read from a sequence ω, the vector $X(t_k)$ will take a particular value X_i. The equation 5.4 proposes an example of a particular value for X_i where the value of x_1 is unknown and then, denoted ϕ. It is is easy to see that such particular value allows to identify the corresponding states in the behavioral model $BM(X(t))$ of figure 5.3.

$$X_{12} = \left\{ \begin{array}{rcl} x_1 & = & \phi \\ x_2 & = & \delta_{22} \\ x_5 & = & \delta_{51} \\ x_6 & = & \delta_{61} \\ x_7 & = & \delta_{71} \\ \phi_8 & = & \delta_{80} \end{array} \right\} \tag{5.4}$$

A sequence of state vector values $\{..., X_i(t_k), ...\}$ is then made with the sequence of values X_i the state vector $X(t)$ take at each time t_k of an occurrence $C_i(t_k)$ returned by the function $read(\omega, t_k)$ of the definition 5.2.

5.6.2 Diagnosing ω_1

According $M(X(t))$, $\omega_1 = \{\ C_{51}(t_2),\ C_{52}(t_3),\ C_{71}(t_4),\ C_{72}(t_5),\ C_{60}(t_7)\}$ leads to:

1. $C_{51}(t_2)$: $x_5(t_2) = \delta_{51}$.
   The initial time t_0 is unknown but less that t_2 (i.e. $t_0 \leq t_2$).
   $R^{ab}_{current} = R^{ab}_{static}$.
   $OM_{78}(X(t))$ contains three labelled observation classes corresponding to C_{51}, $OM_{68}(X(t))$ contains four, but with only one occurrence, no relation can be identified.
   At $t = t_2$, $X(t_2) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{51}, x_6 = \phi, x_7 = \phi, \phi_8 = \phi\}$.

2. $C_{52}(t_3)$: $x_5(t_3) = \delta_{52}$.
   At $t = t_3$, the function $getRelations(C_{51}, C_{52})$ applied on $OM_{78}(X(t))$ returns three relations: r_{712}, r_{722} and r_{752}. Three paths can then be made:
   $P_{78} = \{p_1 = \{r_{712}\}, p_2 = \{r_{722}\}, p_3 = \{r_{752}\}\}$.
   For $X_6(t)$, the same function applied on $OM_{68}(X(t))$ returns also four relations: r_{622}, r_{651}, r_{661} and r_{682}.
   A set P_{68} containing four pathes can be made:
   $P_{68} = \{\ p_4 = \{r_{622}\},\ p_5 = \{r_{651}\},\ p_6 = \{r_{661}\},\ p_7 = \{r_{682}\}\ \}$.
   $X(t_3) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{52}, x_6 = \phi, x_7 = \phi, \phi_8 = \phi\}$. None of these relations is unsatisfactory according to $R^{ab}_{current}$.

3. $C_{71}(t_4)$: $x_7(t_4) = \delta_{71}$.
   At $t = t_3$, $getRelations(C_{52}, C_{71})$ returns only one relation from $OM_{78}(X(t))$: r_{731} which

is satisfactory.
The paths in P_{78} can then be updated according the function $isAPath(p_i, r_{731})$.
For $OM_{78}(X(t))$, $isAPath(p_1, r_{731}) = true$ only for p_1: the other paths can be removed from P_{78}:
$P_{78} = \{p_1 = \{r_{712}, r_{731}\}\}$.
P_{68} is not concerned by $C_{71}(t_4)$.
$X(t_4) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{52}, x_6 = \phi, x_7 = \delta_{71}, \phi_8 = \phi\}$.

4. $C_{72}(t_5)$: $x_7(t_5) = \delta_{72}$.
At $t = t_5$, the goal is reached for the first time: $t_i = t_5$.
$R^{ab}_{current} = R^{ab}_{current} + R^{ab}_{dynamic}$.
The function $getRelations(C_{71}, C_{72})$ returns only one relation, r_{764}, for $OM_{78}(X(t))$ and the function $isAPath(p_1, r_{764})$ returns *true*:
$P_{78} = \{p_1 = \{r_{712}, r_{731}, r_{764}\}\}$.
P_{68} is unchanged.
$X(t_5) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{52}, x_6 = \phi, x_7 = \delta_{72}, \phi_8 = \phi\}$.

5. $C_{60}(t_7)$: $x_6(t_7) = \delta_{60}$.
At $t = t_7$, the function $getRelations(C_{52}, C_{60})$ returns only one relation, r_{632}, for $OM_{68}(X(t))$: $r_{632} \in R^{ab}_{current}$ is not satisfactory. $isAPath(p_i, r_{632})$ returns *true*:
$P_{68} = \{p_4 = \{r_{622}, r_{632}\}\}$.
$X(t_6) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{52}, x_6 = \delta_{60}, x_7 = \delta_{72}, \phi_8 = \phi\}$.
$X(t_6)$ corresponding to an unsatisfactory state, it is necessary to explain $C_{60}(t_7)$.

Providing an explanation to $C_{60}(t_7)$ corresponds to find an *unobserved* occurrence of an observation class that could *entails* $C_{60}(t_7)$ according to $OM_{68}(X(t))$. To this aim, any relation r_n in $OM_{68}(X(t))$ satisfying the relation $r(C_{phi}, C_{60}, \phi)$ so that $isAPath(p_4, r_n)$ returns *true* is *adequate*. According to $OM_{68}(X(t))$, the only relation satisfying these two constraints is $r_{634}(C_{80}, C_{60}, \phi)$: C_{80} corresponds to the proposition $\phi_8(t) = \delta_{80}$. So, r_{634} can be added in p_4: $p_4 = \{\ r_{622},\ r_{632},\ r_{634}\ \}$ and a (unobserved) timed observation $C_{80}(t_k)$ can be added in ω_1 with the temporal constraint $t_2 \leq t_k \leq t_6$.

A look in the graph of figure 5.6 shows that adding $C_{80}(t_k)$ in ω_1 must also lead to add the unobserved occurrence $C_{10}(t_{k+1})$, $t_k \leq t_{k+1} \leq t_6$ by the relation $r_{831}(C_{80}, C_{10}, \phi)$: *the pipe PI is leaking*. A new path $p_8 = \{\ r_{634},\ r_{831}\ \}$ can then be added in P_{68}: $P_{68} = \{\ p_4 = \{\ r_{622},\ r_{632},\ r_{634}\ \},\ p_8 = \{\ r_{634},\ r_{831}\ \}\ \}$. In that case, $X(t_6) = \{x_1 = \delta_{10}, x_2 = \phi, x_5 = \delta_{52}, x_6 = \delta_{60}, x_7 = \delta_{72}, \phi_8 = \delta_{80}\}$.

When replacing the classes with the the corresponding occurences, the class path P_{68} become an *instanciated fault model* as illustrated in the figure 5.8.

It is to note that the relation r_{633} could also be taken under consideration. But this relation only modifies the time t_k of $C_{80}(t_k)$ (i.e. $t_k \leq t_2$ with r_{633}): in most case, experts prefers the most recent time that preceding the fault. Other choices can be done according to the expectations of the diagnosis function of the program $\Theta(X, \Delta)$.

As a conclusion, according to the TOM4D model $M(X(t))$, the following *hypothetical scenario* constitutes a *diagnosis* for $X(t)$ according to ω_1:

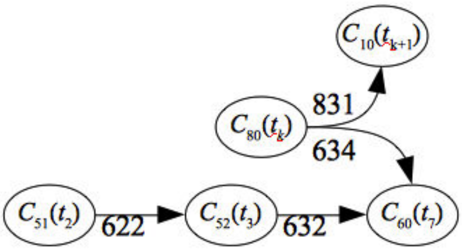


Figure 5.8: An Instanciated Fault Model for ω_1

- Two timed observation $C_{80}(t_k)$, $t_2 \leq t_k \leq t_6$ *explains* why $X(t)$ reached the state $X(t_6) = \{x_1 = \delta_{10}, x_2 = \phi, x_5 = \delta_{52}, x_6 = \delta_{60}, x_7 = \delta_{72}, \phi_8 = \delta_{80}\}$ with ω_1.

- The path P_{68}: $P_{68} = \{\ p_4 = \{\ r_{622},\ r_{632},\ r_{634}\ \},\ p_8 = \{\ r_{634},\ r_{831}\ \}\ \}$ is the corresponding *model of faulty behavior.*

- The instanciated path $IP_{68} = \{\ Ip_4 = \{\ r_{622}(C_{51}(t_2), C_{52}(t_3)),\ r_{632}(C_{52}(t_3), C_{60}(t_7)),\ r_{634}(C_{80}(t_k), C_{60}(t_7))\ \},\ Ip_8 = \{\ r_{634}(C_{80}(t_k), C_{60}(t_7)),\ r_{831}(C_{80}(t_k), C_{10}(t_{k+1}))\ \}\ \}$ is the corresponding *instanciated faulty model* that *describes* the way the process reached $X(t_6)$ (cf. 5.8).

This diagnosis is coherent with the initial senario which contains the timed observation $C_{10}(t_6)$: $t_{k_8} \leq t_6 \leq t_7$. But it proposes also an *explanation* for $C_{10}(t_6)$: $\phi_8 = \delta_{80}$. In other words, according to this hypothesis, the *pipe PI is leaking* because the surface of the *pipe PI* is not sufficient to conduct the output flow of the *pump P* in the *tank TA*: may be the pipe is porous, may be there is a hole, may be $flow_p$ is too high for the pipe. Only a physical observation can transforms this hypothesis in a fact.

5.6.3 Diagnosing ω_2

According $OM(X(t))$, $\omega_2 = \{\ C_{51}(t_{11}),\ C_{52}(t_{12}),\ C_{71}(t_{13}),\ C_{72}(t_{14}),\ C_{51}(t_{16}),\ C_{50}(t_{18})\}$ leads to:

1. $C_{51}(t_{11})$: $x_5(t_{11}) = \delta_{51}$.
   $t_0 \leq t_{11}$.
   $R^{ab}_{current} = R^{ab}_{static}$.
   $X(t_{11}) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{51}, x_6 = \phi, x_7 = \phi, \phi_8 = \phi\}$.

2. $C_{52}(t_{12})$: $x_5(t_{12}) = \delta_{52}$.
   $P_{78} = \{p_1 = \{r_{712}\}, p_2 = \{r_{722}\}, p_3 = \{r_{752}\}\}$.
   $P_{68} = \{\ p_4 = \{r_{622}\},\ p_5 = \{r_{651}\},\ p_6 = \{r_{661}\},\ p_7 = \{r_{682}\}\ \}$.
   $X(t_{12}) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{52}, x_6 = \phi, x_7 = \phi, \phi_8 = \phi\}$.

3. $C_{71}(t_{13})$: $x_7(t_{13}) = \delta_{71}$.
   $P_{78} = \{p_1 = \{r_{712}, r_{731}\}\}$.
   $P_{68} = \{\ p_4 = \{r_{622}\},\ p_5 = \{r_{651}\},\ p_6 = \{r_{661}\},\ p_7 = \{r_{682}\}\ \}$ (no change).
   $X(t_{13}) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{52}, x_6 = \phi, x_7 = \delta_{71}, \phi_8 = \phi\}$.

4. $C_{72}(t_{14})$: $x_7(t_{14}) = \delta_{72}$.
   $t_i = t_{14}$.

$R^{ab}_{current} = R^{ab}_{current} + R^{ab}_{dynamic}$.
$P_{78} = \{p_1 = \{r_{712}, r_{731}, r_{764}\}\}$.
$P_{68} = \{\ p_4 = \{r_{622}\},\ p_5 = \{r_{651}\},\ p_6 = \{r_{661}\},\ p_7 = \{r_{682}\}\ \}$ (no change).
$X(t_{14}) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{52}, x_6 = \phi, x_7 = \delta_{72}, \phi_8 = \phi\}$.

5. $C_{51}(t_{16})$: $x_5(t_{16}) = \delta_{51}$.
   $P_{78} = \{p_1 = \{r_{712}, r_{731}, r_{764}, r_{782}\}\}$.
   $P_{68} = \{\ p_4 = \{r_{622}, r_{631}\},\ p_5 = \{r_{651}, r_{643}\},\ p_6 = \{r_{661}, r_{671}\},\ p_7 = \{r_{682}, r_{691}\}\ \}$.
   $X(t_{16}) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{51}, x_6 = \phi, x_7 = \delta_{72}, \phi_8 = \phi\}$.

6. $C_{50}(t_{18})$: $x_5(t_{18}) = \delta_{50}$.
   $P_{78} = \{p_1 = \{r_{712}, r_{731}, r_{764}, r_{782}, r_{751}\}\}$.
   $P_{68} = \{\ p_4 = \{r_{622}, r_{631}, r_{625}\},\ p_7 = \{r_{682}, r_{691}, r_{683}\}\ \}$.
   $X(t_{18}) = \{x_1 = \phi, x_2 = \phi, x_5 = \delta_{50}, x_6 = \phi, x_7 = \delta_{72}, \phi_8 = \phi\}$.

At the end of ω_2, X_7 is in the state s_{77} (cf. figure 5.3) and there is an ambiguity about X_6 that can be either in the states s_{60} or s_{61}. But the main problem is that the operating goal is reached since t_{14}, but no unsatisfactory states has been detected during ω_2.

Nevertheless, it is possible to try to get a diagnosis at the end of ω_2 when finding any *unobserved* occurrence that could *explain* ω_2. Applied to $OM(X_{78})$, the functions $getRelations(C_\phi, C_{50})$ provides only one relation $r_{803}(C_{20}, C_{50}, [\tau^{-}_{803}, \tau^{-}_{803}])$ making a relation between $C_{50}(t_{18})$ and an *unobservable* class C_{20} (no relation for $OM(X_{68})$). An (unobserved) occurrence $C_{20}(t_k)$ can be added in ω_2 with $t_k \leq t_{18}$. The path $P_{78} = \{\ p_1 = \{\ r_{712},\ r_{731},\ r_{764},\ r_{782},\ r_{751},\ r_{803}\ \}\ \}$ can then be proposed as a *model of fault* for ω_2, its corresponding *instanciated fault model* being $IP_{78} = \{\ Ip_1 = \{\ r_{712}(C_{51}(t_{11}), C_{52}(t_{12})),\ r_{731}(C_{52}(t_{12}), C_{71}(t_{13})),\ r_{764}(C_{71}(t_{13}), C_{72}(t_{14})),\ r_{782}(C_{72}(t_{14}), C_{51}(t_{16})),\ r_{751}(C_{51}(t_{16}), C_{50}(t_{18})),\ r_{803}(C_{20}(t_k), C_{50}(t_{18}))\ \}\ \}$ (cf. figure 5.9).

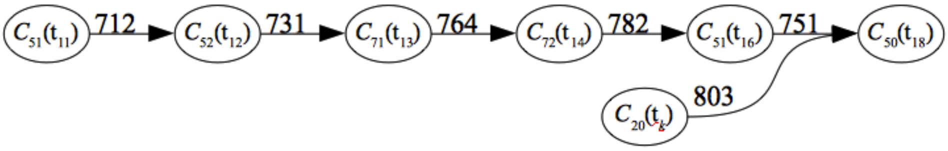


Figure 5.9: An Instanciated Fault Model for ω_2

The occurrence $C_{20}(t_k)$ (i.e. $x_2(t_k) = \delta_{20}$ means that the *pump P is blocked*: this *hypothetical scenario* is then coherent with the initial scenario which contains the occurrence $C_{20}(t_17)$.

It is to note that doing the same reasoning with $\omega_2 - C_{50}(t_{18})$ allows also to find a diagnosis: an occurrence $C_{21}(t_{k-1})$ (i.e. the *pump P is leaking*) would be added which is coherent with the occurrence $C_{21}(t_{15})$. Inversely, supposing that ω_2 contains the two other occurrences $C_{71}(t_{19})$ and $C_{70}(t_{20})$, the same reasoning in the past would lead to the same conclusions. This shows an important property of the proposed diagnosis algorithm: it can be interrupted at any time. If there exists at least a diagnosis, it will provide it.

It is to note that the values of the state vector $X(t_k)$ play no role in the algorithm proposed in the next section: it can be usefull for the human interpretations but no decisions are based on it.

5.7 A Timed Observation Driven Diagnosis Algorithm

The proposed algorithm is called TOM4E which means *Timed Observation Management for Explanation.*

This aim of TOM4E is to produce an explanation to each unsatisfactory relation $r_k($ $C_p(t_k)$, $C_s(t_{k+1})$, $[\tau^-, \tau^+])$ a given occurrence $C_i(t_k)$ allows to detect at time t_k. To this aim, TOM4E execute systematically the same work as the one made to diagnose ω_1 and ω_2. For each occurrence $C_i(t_k)$ written by an observation program $\Theta(X, \Delta)$, TOM4E execute the following tasks on each observation model $OM(X_i(t))$ linked with a process $X(t) = \bigcup X_i(t)$:

- $detect(r, i)$
- $describe(r, i)$
- $explain(r, i)$

The result is a $E = \{E_i\}$ of *explanations* that are provided under the form of pairs $E_i = (EP_i, R_i^{ab})$ where :

- $EP_i = \{ep_k\}$ is a (possibly empty) set of instantiated paths $ep_k = \{..., r_k(C_p(t_k), C_s(t_{k+1}), [\tau^-, \tau^+])$ $, ...\}$,
- each path ep_k is made with a series of instantiated relation of the form $r_k(C_p(t_k), C_s(t_{k+1}), [\tau^-, \tau^+]$,
- eventually, unobservable occurrences, constituting so many *explanations* of EP_i, have been used to completed the series,
- each EP_i is associated with the corresponding set R_i^{ab} of unsatisfactory relations $r_n(C_p, C_s, [\tau^-, \tau^+])$ that *justify* the explanations.

In other words, each paths EP_i is a *instantiated faulty models* for a process $X_i(t)$ according to an observation model $OM(X_i(t))$. For example, the set E contains the instantiated faulty model of figure 5.10 produced by TOM4E at the end of ω_1 with $OM(X_{78}(t))$ and $OM(X_{68}(t))$ in the TOM4E knowledge base.

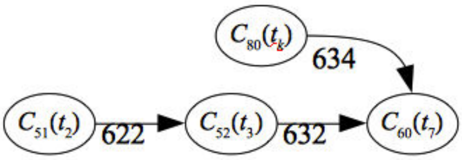


Figure 5.10: TOM4E Instantiated Fault Model for ω_1

Because $X_7(t)$ is in a satisfactory state at the end of ω_2, TOM4E provides no explanations (i.e. E is empty). This clearly comes from the definition of the operating goal: *a priori*, there is no reason to look for an explanation at the end of ω_2: the pump can have been turned off because the tank TA is sufficiently full. To improve the results of TOM4E with ω_2, the best thing to do is to define more precisely the operating goal of $X(t)$.

But, on another hand, it is easy to see that an additional function can be implemented to *force* TOM4E to look for an explanation even when the state is satisfactory. For example, the $provideAnExplanation(i)$ function in the suite of the section forces TOM4E to provide the explanation of the figure 5.9 at the end of ω_2. It is also easy to add a temporal depth, under the form of an integer n, wich aims at forcing the explanation up to the nth latter occurrence. For example, with $n = 2$, TOM4E would provides the explanation of the figure 5.11 where the explanation $r_{803}(C_{21}(t_{k-1}, C_{51}(t_{16})$ has been added.

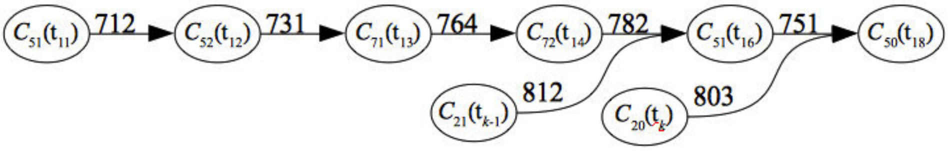


Figure 5.11: Instantiated Fault Model for ω_2 made by a $provideAnExplanation(i, 2)$ Function

The TOM4E algorithm is clearly designed to be *Timed Observation Driven*: this means that when the program $\Theta(X, \Delta)$ writes an occurrence $C_{ij}(t_k)$ in a sequence ω (i.e. executes $write(\omega, C_{ij}(t_k)))$, the algorithm TOM4E can be executed *on the flow* to produce a set E of instantiated fault models corresponding to ω. In other words, the TOM4E algorithm is said to be *Any Time*: when interrupted, it alway provides a solution E, even empty, to the diagnosis problem. The *Any Time* property of TOM4E explains why no explanation is provided by TOM4E with ω_2: this is the result of a design decision. Nevertheless, the function $provideAnExplanation(i)$ shows that it is possible to complete the algorithmic properties of a diagnosis system implementing the TOM4E algorithm.

The algorithms and functions required to implement the TOM4E algorithm are proposed at the end of this section.

5.8 Conclusion

This chapter proposes the TOM4E algorithm to diagnose dynamic systems modeled with the TOM4D methodology according to the Timed observations Theory of [LG06].

In practice, these algorithms and functions have been implemented in the Java tool called *ELP Lab* that has designed and developed by Pr. Le Goc at LSIS, the Laboratory of Sciences of Information and Systems of Aix-Marseille University (France). This software is designed as a Java environment allowing to integrate and to test various TOT based algorithms aiming at:

- discovering temporal knowledge (i.e. sets of timed binary relations) from timed data,
- modeling these sets as observation models according to the definition 5.3 that are graphically displayed as graphs to visualize these sets,
- validating the models with timed data and using the models to the monitoring, the diagnosis and the prognosis of dynamic systems.

This algorithm has been implemented with a generator of *abstract binary observers networks* (cf. [LG06]) that transforms a set R of timed binary relation $r_k(C_p(t_k), C_s(t_{k+1}), [\tau^-, \tau^+])$ of an observation model $OM(X(t))$ into a network of *abstract binary observers* that produces the same result as TOM4E. The general principle consists in building a network for an observation model $OM(X(t))$ where one instance of an *abstract binary observer* is associated to each timed binary relation $r_k(C_p(t_k), C_s(t_{k+1}), [\tau^-, \tau^+])$ of $OM(X(t))$. A *distributor* distributes the occurrences $C_i(t_k)$ directly to the *abstract binary observers* that are concerned with, and the observers sends the expected results to a *collector* managing the set E. In other words, the *Detect-Describe-Explain* main functions of TOM4E are distributed on each individual *abstract binary observers* of the two networks implementing $OM(X_{78})$ and $OM(X_{68})$ of the example.

The advantage is the *abstract binary observers* technology is that such a network realizes a purely *time driven* and *distributed* computation. As a consequence, the computation of the TOM4E algorithm is very fast and parsimonious in memory. This means that is can easily be used in real time, even with very large scale networks of dynamic processes. The TOM4E algorithm being *Any Time*, with the *abstract binary observers* technology it can also be used *Real Time*.

Yet, the next chapter illustrates the application of the TOM4E algorithm on a real-world hydraulic dynamic process, the Sapin's dam.

Algorithm: $TOM4E(C_i(t_k))$
Inputs:
An occurrence $C_j(t_k)$ of an observation class C_i.
Outputs:
A sorted set $E = \{E_i\}$ of n pairs $E_i = (EP_i, R_i^{ab})$ linking a possibly empty set
$EP_i = \{ep_k\}$ of instantiated paths.
$ep_k = \{..., r_k(C_p(t_k), C_s(t_{k+1}), [\tau^-, \tau^+], ...\}$ made with a series
of instantiated relation of the form $r_k(C_p(t_k), C_s(t_{k+1}), [\tau^-, \tau^+])$
eventually completed with not observed occurrences constituting an explanation
with the corresponding set R_i^{ab} of unsatisfactory relations $r_n(C_p, C_s, [\tau^-, \tau^+])$.

Knowledge Base:
The set $R = \{R_i\}$ of the sets R_i of timed binary relation of n
observation models $OM(X_i(t))$.
The set $R^{ab}_{current} = \bigcup R^{ab}_{static\_i}$ of the union set of the $R^{ab}_{static}(i)$
of the n observation models $OM(X_i(t))$.
The set $R^{ab}_{dynamic} = \{R^{ab}_{dynamic\_i}\}$ of the n sets $R^{ab}_{dynamic}$
of the observation models $OM(X_i(t))$.
The set $R^u = \bigcup R^u_i\}$ of the n sets R^u_i containing the unobservable relations
of the observation models $OM(X_i(t))$.

Long term memory:
The initial time stamp t_0 initialized with the value -1.
The precedent and the current occurrence $C_i(t_k - 1)$ and $C_j(t_k)$ initialized with the value ϕ.
The set $T = \{t_i\}$ of n time stamps $t(i)$ at which a process $X_i(t)$ reached its goal,
each t_i being initialized withe the value -1.
The set $P = \{P_i\}$ of n sets P_i of paths p_k of relations: $p_k = \{..., r_k((C_p, C_s, [\tau^-, \tau^+]), ...\}$,
The set $I = \{IP_i\}$ of n sets IP_i of paths ip_k of instantiated relations:
$ip_k = \{..., r_k(C_p(t_k), C_s(t_{k+1}), [\tau^-, \tau^+], ...\}$.
The set $E = \{EP_i\}$ of sets EP_i of paths ep_k of instantiated relations: .
$ep_k = \{..., r_k(C_p(t_k), C_s(t_{k+1}), [\tau^-, \tau^+], ...\}$
containing unobserved occurrences $C_e(t_e)$ that could explain a instantiated relation r
All these sets of paths are initialized as an empty set denoted Φ.
The set $S = \{s_i\}$ of n boolean variables the value of which is TRUE when a relation r
of the form $r(C_iC_j, [\tau^-, \tau^+])$ is satisfactory according to $OM(X(t))$.
All the s_i of S are initialized with a TRUE value.

```
1. IF $t_0 = -1$ THEN $t_0 = t_k$
2. IF $C_j(t_k - 1) = \phi$ THEN
2.1 $C_j(t_k - 1) = C_i(t_k)$
2.2 return $E$
//A binary relation can be made with $C_j(t_k - 1)$ and $C_i(t_k)$
//Detect, describe and explain $C_i(t_k)$ for each $OM(X_i(t))$
3. $\forall i = 1...n$
//Get all the relations in $R(i)$ having the form $r_n(C_iC_j, [\tau^-, \tau^+])$
3.1. $rSet = getRelations(C_i, C_j, R(i))$
3.2. $\forall j = 1...rSet.size()$
3.2.1. $r = rSet(i)$
3.2.2. $detect(r, i)$
3.2.3. $describe(r, i)$
3.2.4. $explain(r, i)$
4. $C_i(t_k - 1) = C_j(t_k)$
5. return $E$
```

```
Algorithm: $detect(r, i)$
Inputs:
The index $i$ of the set $R(i)$ containing the relations of $OM(X_i(t))$ to consider.
A binary relation $r_n(P, S, [\tau^-, \tau^+])$.
Outputs:
The set $P(i)$ of $P$ containing $n_{p_i}$ paths $p$ of relations:
$p = \{..., r_k((C_p, C_s, [\tau^-, \tau^+]), ...\}$.

1. IF $isSatisfactory(r, R^{ab}_{current}(i)) = true$ THEN
// update $t_i$ and $R^{ab}_{current}(i)$
1.1 $T(i) = t_k$
1.2 $R^{ab}_{current}(i) = R^{ab}_{current}(i) + R^{ab}_{dynamic}(i)$
1.3 $S(i) = $ TRUE
2. ELSE //update $R^{ab}_{current}(i)$ only
2.1 $S(i) = $ FALSE
2.2. $R^{ab}_{current}(i) = R^{ab}_{current}(i) - R^{ab}_{dynamic}(i)$
//If $P(i)$ is empty, create a new path containing r and add it in $P(i)$
3. IF $P(i) = \Phi$ THEN
1.1. $p = \{r\}$
1.2. $P(i) = P(i) + p$ 1.3. return
// $P(i)$ is not empty
4. $\forall k = 1...P(i).size()$ //For all the pathes $p \in P(i)$
4.1. $p = P(i)(k)$
4.2. IF $isAPath(p, r, R(i))$ THEN $p = p + r$
4.3. ELSE $P(i) = P(i) - p$ //p is no more a path for $OM(X_i(t))$
5. return
```

```
Algorithm: $describe(C_i(t_k), r, i)$
Inputs:
The index $i$ of $OM(X_i(t))$.
A binary relation $r_n(P, S, [\tau^-, \tau^+])$.
Outputs:
The set $IP(i)$ of $I$.

1. $\forall k = 1...IP(i).size()$ //For all the paths $p \in P(i)$
//If the path $P(i)(k)$ has been removed from $P(i)$
1.1. IF $P(i)(k) = \Phi$ THEN $IP(i)(k) = \Phi$
1.2. ELSE
//Create an instantiated version of $r$
1.2.1 $ir = ir_n(C_j(t_{k-1}), C_i(t_k), [\tau^-, \tau^+])$
//Add $ir$ in $IP(i)(k)$
1.2.2 $IP(i)(k) = P(i)(k) + ir$
3. return
```

Algorithm: $explain(r, i)$
Inputs:
The index i of $OM(X_i(t))$.
A binary relation $r_n(P, S, [\tau^-, \tau^+])$.
Outputs:
The set $EP(i)$ of E.

```
1. $\forall k = 1...EP(i).size()$ //For all the paths $p \in EP(i)$
    //The path $P(i)(k)$ has been removed from $P(i) \Rightarrow$ clear $EP(i)(k)$
    1.1. IF $P(i)(k) = \Phi$ THEN $EP(i)(k) = \Phi$
    1.2. IF $S(i) =$ FALSE THEN
        //Find the set of relations of the form $r_e(C_e, P, [\tau_m^-, \tau_m^+])$ in $R(i)$
        1.2.1. $eSet = getExplanations(r, R(i))$
        //Add $eSet$ in $EP(i)(k)$
        1.2.2. $EP(i)(k) = EP(i)(k) + eSet$
2. return
```

Algorithm: $getExplanations(r, R)$
Inputs:
A binary relation $r_n(P, S, [\tau^-, \tau^+])$.
The set R of an observation model $OM(X_i(t))$.
Output:
A set E of binary relation r_{ps} of the form $r_k(e, P, [\tau_{ps}^-, \tau_{ps}^+])$ $s \subseteq R$.

```
1. $E = \Phi$
2. $\forall i = 1...R.size()$
    2.1. $rR = R(i)$
    2.3. $sR = r.getS()$
    2.4. IF $r.getS() = rR.getP()$ THEN $E = E + r$
//Remove the observables relations
3. $\forall i = 1...E.size()$
    3.1. $e = E(i)$
    3.2. $found =$ FALSE
    3.3. $\forall j = 1...R^u.size()$
        3.3.1. $u = R^u(j)$
        3.3.2. IF $found =$ FALSE OR $e = u$ THEN $found =$ TRUE
    3.4. IF $found =$ FALSE THEN $E = E - e$
4. return E
```

Algorithm: $provideAnExplanation(i)$
Inputs:
The index i of an observation model $OM(X_i(t))$.
Output:
A set E of binary relation r_{ps} of the form $r_k(e, P, [\tau_{ps}^-, \tau_{ps}^+])$ $s \subseteq R$.

```
1. $E = E(i)$
2. $\forall i = 1...E.size()$
    2.1. $e = E(i)$
    2.2. $r = e.getLastRelation()$
    2.3. $getExplanations(r, R(i))$
4. return E
```

CHAPTER 6

Application to the Hydraulic Dam Safety Assessment

Dams are heterogeneous structures featured by complex behaviors that evolve through time because of their natural aging. This aging can be accelerated by environmental causes (climatic conditions, floods and earthquakes) or by internal causes (poor design or construction conditions, insufficient or inadequate maintenance...). These causes involve, during the life of the structure, the occurrence and the development of deterioration phenomena, more or less dependent and stemming from miscellaneous and complex sources. Such phenomena are for instance, fissure of the concrete facing, clogging of the drain outlet, sliding of shoulders [CIG94]. Figure 6.1 illustrates two examples of deterioration phenomena. The first photography (a) shows the deterioration of the protection of the upstream facing: cobblestones are lacking and the geotextile underneath is visible and locally deteriorated. The second photography (b) shows the downstream shoulder of an embankment dam on which the vegetation is composed of shrubs and young trees. The death of this type of vegetation can generate specific water circulation caused by the disappearance of roots.

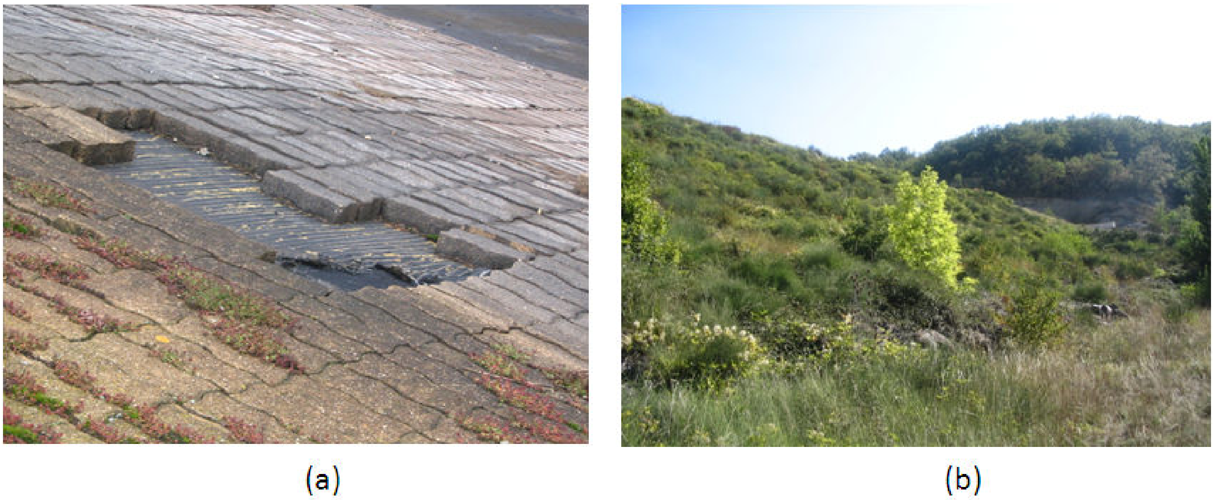


Figure 6.1: Examples of deterioration phenomena - Photos Irstea

Some phenomena can lead to the failure of the structure such as what happened to the Malpasset Dam in France in 1959 (cf. Figure 6.2).

At the moment, throughout the world, the assessment of dam reliability and safety, their diagnosis and the proposal for corrective actions are carried out by expert engineers, during dam

Figure 6.2: Failure of the Malpasset Dam (France) in 1959 - Photos Irstea

reviews. Indeed, it is crucial to develop methods and tools for managing the dynamic behavior of dams and modeling the process.

This chapter apply the TOM4D methodology and the TOM4E algorithm described in the previous sections to the French Sapins dam presented in Section 6.1. Sections 6.2, 6.4 and 6.5 present the application of the TOM4D methodology to the Sapins' dam modeling. Section 6.6 focalizes on the application of the diagnosis algorithm defined in previous chapter with the TOM4D model of this dam. Finally, sections 6.7 concludes this chapter with an interpretation of the obtained results.

6.1 Sapins dam presentation

The phenomena that occurred during the life of the French Sapins Dam are described in [PRB06]. This knowledge base comprises text and several figures.

Let us consider the available knowledge extracted from [PRB06] where only the internal erosion scenario is considered:

1. The Sapins dam is a homogeneous dam, 16 m high and impounding a 2 hm^3 lake.
2. this dam is set up by a upstream shoulder and a downstream shoulder composed of granite arena structure based on a granite foundation. It comprises a vertical drain between the upstream and the downstream shoulder and a horizontal drain at the interface of the

foundation with the downstream half of the dam (Figure 6.3). The horizontal drain is considered to drain only the foundation and the vertical one, only the upstream shoulder. The top of the vertical drain is 2 m lower than the normal reservoir level.

3. Sensors and visual observations are available:
   - the water level of the reservoir is assessed through a depth gauge;
   - the upstream shoulder is instrumented with three pore pressure cells C_3, C_4 and C_5 that assess the water level in the component;
   - the vertical drain discharge is recorded;
   - wet areas and leakages can be observed on the downstream shoulder.

Wet areas and leakages are observed by human experts and so, considered as absract sensors (such observations are considered as *indicators* [CPB10] that provide a reliable assessment).

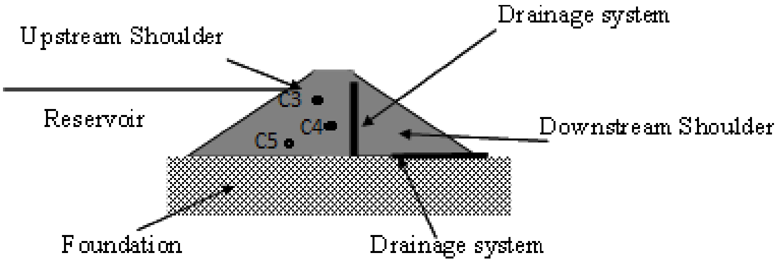


Figure 6.3: Structure of the Sapins dam

The knowledge base allows a chronological analysis of the events that marked the life of this dam on the basis of the data stemming from monitoring devices and visual observations, after the construction of the Sapins Dam that ended in November 1978 (cf. Figure 6.4):

1. In November 1978, the reservoir filling began and the pore pressure cells stabilized at normal values

2. In December 1978, the reservoir got its normal water level elevation

3. In April 1979, the pressures assessed by $C4$ and $C5$ in the upstream shoulder indicated values higher than the normal (waited for) ones

4. In November 1980, the drainage flow gets a normal and progressive raise

5. In November 1980, a decrease of this flow was assessed

6. In November 1981, a partial emptying of the reservoir was performed

7. In November 1981, the drainage flow increased

8. In December 1981, the reservoir was again at its normal level elevation

9. In October 1982, the drainage flow decreased again (the low level is obvious on February 1983)

10. In December 1985, a new partial emptying of the reservoir was carried out

11. In March 1986, the reservoir was again at its normal level

12. In September 1988, a very wet area was noted at the toe of the downstream shoulder

13. In mid-October 1988, muddy seepage could be observed on the downstream shoulder

For safety reasons, it was decided to completely empty the reservoir and improve its structural safety before bringing the dam back into service.

These elements of description are given with the purpose of a diagnosis cognitive task. We propose to use the TOM4D formalism along with the algorithms developed to explain the phenomena that occured during 10 years in the Sapins Dam.

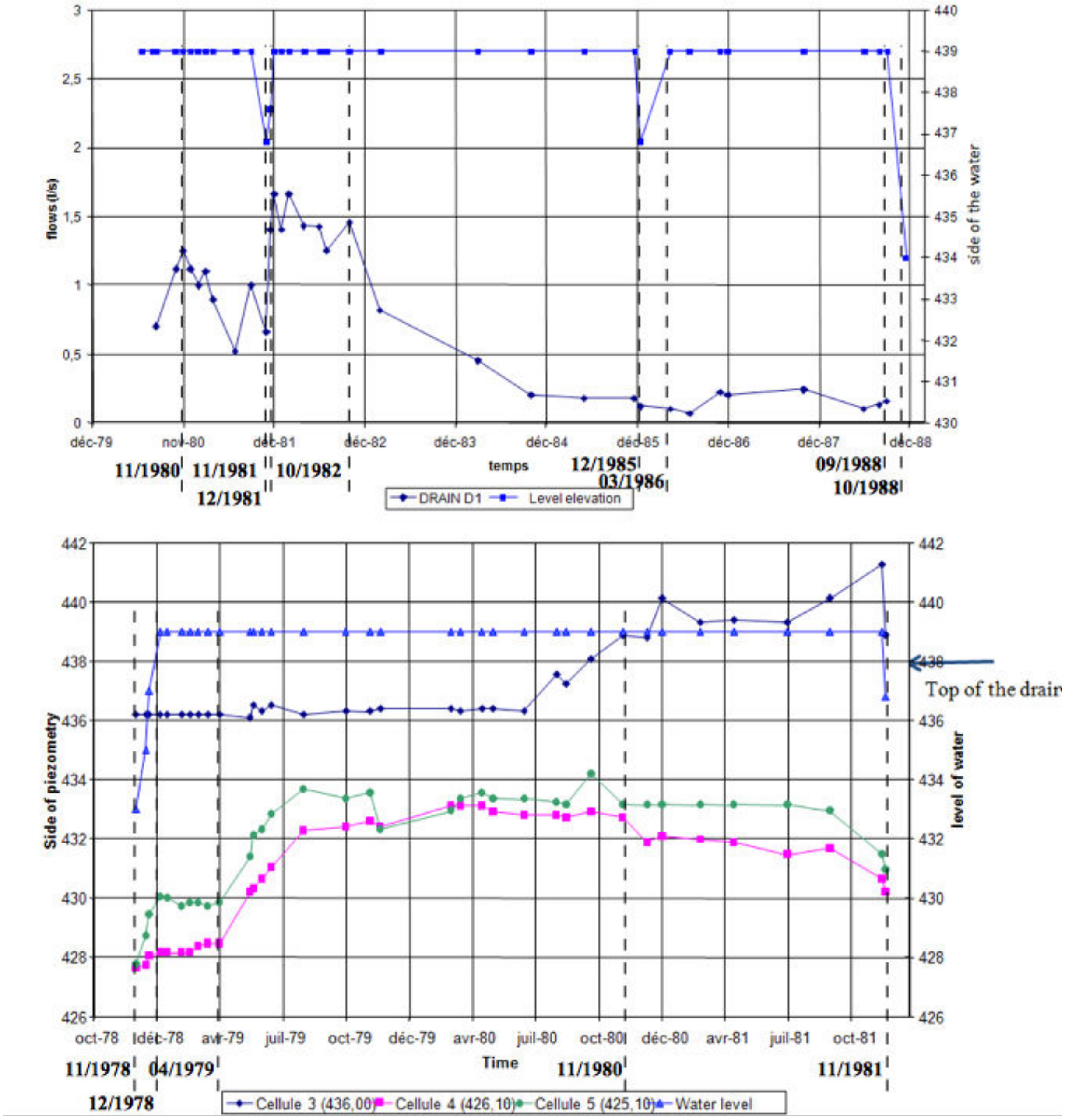


Figure 6.4: Sapins'Dam Evolution through Piezometry, Drain Outflow and Reservoir Level [PRB06]

6.2 Knowledge Interpretation

A first knowledge interpretation of the knowledge base extracted from [PRB06].

This interpretation comes from works developed in [LGM07] and [LGMC08] but in this chapter, the *Dam Crest* is considered as included in the *UpShoulder* and the *DownShoulder*, and the *anchor trench* is considered as a part of *Foundation.*

Six components and five sensors are explicitely defined as *components*:

- Component c_1: The name of this component is *Vertical Drain.* It is associated with the sensor *DrainFlow* c_7;

- Component c_2: The name of this component is *Reservoir.* It is associated with the sensor *ResLevel* c_8;

- Component c_3: The name of this component is *Upstream Shoulder.* It is associated with the sensors $C1$, $C2$ and $C3$ that assess the pezometry in c_3 during time with values evolving on a continuous scale. Assessing the same *Piezometry* variable (the difference between them comes from their variation range, cf. Figure 6.4), these sensors are combined in a conglomerate variable called *PressureCell* defined on a set of three values: *Medium*, *Low* and *High*;

- Component c_4: The name of this component is *Downstream Shoulder.* It is associated with the visual observations *Leakage* and *WetArea* (c_{10}) that assess the presence of clear or muddy water in c_4 during time;

- Component c_5: The name of c_5 is *Foundation.* There is no sensor associated with it;

- Component c_6: The name of c_6 is *Horizontal Drain.* There is no sensor associated with it.

- Component c_7: The name of c_7 is $DrainFlow$. This sensor is associated with the *Vertical Drain* c_1. The measures over time evolves on a continuous scale that is discretized in three values: *Medium*, *Low* and *High*;

- Component c_8: The name of c_8 is $ResLevel$. This sensor assesses the water level in the *Reservoir* c_2. The experts descretizes this level with three discrete values: *Medium*, *Low* and *High*;

- Component c_9: The name of this component is $PressureCell$. It provides continuous values that are discretized by experts as *Medium*, *Low* and *High*;

- Component c_{10}: This component is an abstract sensor representing the human assessment about two kinds of visual observations, *Leakage* and *WetArea.*

  - *Leakage.* Three possible values are considered: *AbsLeak* (Absence of Leakage in the Downstream Shoulder), *PresLeak* (Presence of Leakage in the Downstream Shoulder) and *SuspLeak* (Suspicion of Leakage in the Downstream Shoulder) ;
  - *WetArea.* Three possible values are also considered: *AbsWA* (Absence of Wet Area in the Downstream Shoulder), *SuspWA* (Suspicion of Wet Area in the Downstream Shoulder) and *PresWA* (Presence of Wet Area in the Downstream Shoulder).

It is to note that only the values *PresLeak* and *PresWA* can be observed on the Downstream Shoulder c_4. The values *SuspWA* and *SuspLeak* means then that even in absence of observable presence of leakage or wet areas, water can be present in the Downstream Shoulder c_4 even if not visible on the slope. If the suspicion is confirmed, one can be sure that there is a leakage (wet area) but it can be not already visible on the downstream slope. The no detection of *Leakage* or *Water Areas* does not mean that there is absolutely no water in the downstream shoulder c_4 but it simply means either there is effectively no water either there is water but it has not reached the slope.

According to the TOM4D methodology, these elements leads to the construction of Table 6.1 where abstract denominations are used to define the variables and their possible values.

C	**X**	**Definition Domain**	**Variable Name**	**Respective Value Name**
c_7	x_7	$\Delta_{x_7} = \{\delta_{70}, \delta_{71}, \delta_{72}\}$	$DrainFlow$	$Low, Medium, High$
c_8	x_8	$\Delta_{x_8} = \{\delta_{80}, \delta_{81}, \delta_{82}\}$	$ResLevel$	$Low, Medium, High$
c_9	x_9	$\Delta_{x_9} = \{\delta_{90}, \delta_{91}, \delta_{92}\}$	$PressureCell$	$Low, Medium, High$
c_{10}	x_{10}	$\Delta_{x_{10}} = \{\delta_{100}, \delta_{101}, \delta_{102}\}$	$Leakage$	$PresLeak, SuspLeak, AbsLeak$
c_{10}	x_{11}	$\Delta_{x_{11}} = \{\delta_{110}, \delta_{111}, \delta_{112}\}$	$WetArea$	$PresWA, SuspWA, AbsWA$

Table 6.1: Definition of the Abstract Variables for the Sapins Dam

Moreover, the application of the *Spatial Discretization Principle* allows determine the definition domains of the sensor's variables: ten abstract thresholds are identified (cf. Table 6.2).

Variable	**Definition Domain**	**Respective Abstract Ranges**
x_7	$\Delta_{x_7} = \{\delta_{70}, \delta_{71}, \delta_{72}\}$	$]-\infty, \Psi_{71}[,\ [\Psi_{71}, \Psi_{72}[\ ,\ [\Psi_{72}, +\infty[$
x_8	$\Delta_{x_8} = \{\delta_{80}, \delta_{81}, \delta_{82}\}$	$]-\infty, \Psi_{81}[,\ [\Psi_{81}, \Psi_{82}[\ ,\ [\Psi_{82}, +\infty[$
x_9	$\Delta_{x_9} = \{\delta_{90}, \delta_{91}, \delta_{92}\}$	$]-\infty, \Psi_{91}[,\ [\Psi_{91}, \Psi_{92}[\ ,\ [\Psi_{92}, +\infty[$
x_{10}	$\Delta_{x_{10}} = \{\delta_{100}, \delta_{101}\}$	$]-\infty, \Psi_{101}[,\ [\Psi_{101}, \Psi_{102}[\ ,\ [\Psi_{102}, +\infty[$
x_{11}	$\Delta_{x_{11}} = \{\delta_{110}, \delta_{111}\}$	$]-\infty, \Psi_{111}[,\ [\Psi_{111}, \Psi_{112}[\ ,\ [\Psi_{112}, +\infty[$

Table 6.2: Definition of The Abstract Thresholds for the Sapins Dam

6.3 Scenario Model

The interpretation of the text is sufficient to define a first structural model $SM(\Omega)$, before providing the Ω sequence (cf. figure 6.5).

Formally, according to the definition 4.2 of the TOM4D methodology, the structural model $SM(\Omega)$ is a structure $< COMPS, R^p, R^x >$ where:

- $COMPS = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}$
- $R^p = \{out_1(c_2) = in(c_8), out_2(c_2) = in(c_3), out_3(c_2) = in(c_5), out_1(c_3) = in(c_9), out_2(c_3) = in(c_1), out_3(c_3) = in_1(c_4), out_1(c_1) = in(c_7), out_2(c_1) = in_2(c_4), out(c_4) = in(c_{10}), out(c_5) = in(c_6), out(c_6) = in(c_4)\}$
- $R^x = \{x_7 = out(c_7), x_8 = out(c_8), x_9 = out(c_9), x_{10} = out_1(c_{10}), x_{11} = out_2(c_{10})\}$

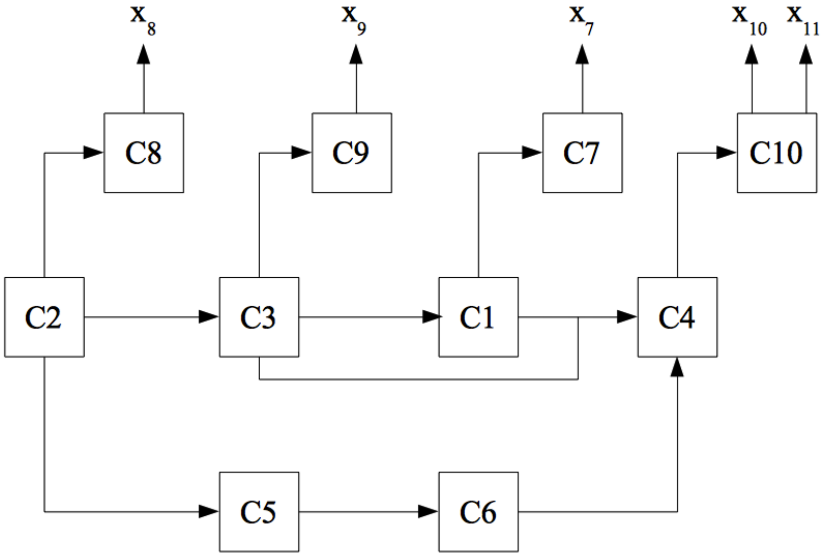


Figure 6.5: Structural Model $SM(\Omega)$ - Sapins Dam

A first structural model being made, it is then possible to analyse the relations between the values of the variables. The structural model of figure 6.5 allows the identification of the following functions (cf. figure 6.6):

- $x_9 = f_1(x_8)$: there is a relation between the values of x_9 and those of x_8;
- $x_7 = f_2(x_9)$: there is a relation between the values of x_7 and those of x_9;
- $x_{10} = f_3(x_9, x_7)$ and $x_{11} = f_4(x_9, x_7)$: there is a relation between the values of x_{10} (resp. x_{11}) and those of x_9 and x_7.

It is to note that physically, the water detected in the downstream shoulder c_4 comes from the vertical drain c_1, the horizontal drain c_6 or from the upstream shoulder c_3. Because no variable is associated with the horizontal drain c_6, we considered that water can only come from the upstream shoulder c_3 otherwise the vertical drain c_1.

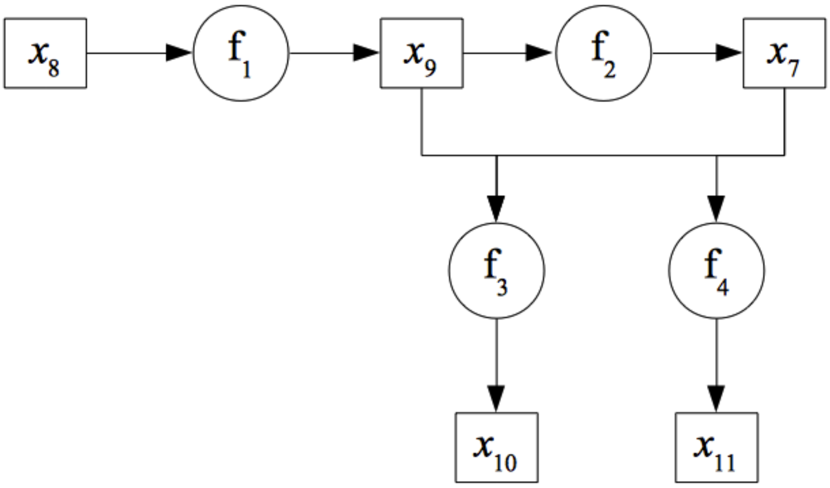


Figure 6.6: Functional Model $FM(\Omega)$

Formally, according to the definition 4.3 of the TOM4D methodology, the functional model $FM(\Omega)$ is a structure $< \Delta, F, R^f >$ where:

- $\Delta = \underset{i=7,8,9,10,11}{\cup} \Delta_{x_i}$ (the Δ_{x_i} are given in the table 6.1).
- $F = \{\ f_1 : \Delta_{x_8} \to \Delta_{x_9},\ f_2 : \Delta_{x_9} \to \Delta_{x_7},\ f_3 : \Delta_{x_9} \times \Delta_{x_7} \to \Delta_{x_{10}},\ f_4 : \Delta_{x_9} \times \Delta_{x_7} \to \Delta_{x_{11}}\ \}$.
- $R^f = \{\ x_8 = f_1(x_9),\ x_7 = f_2(x_9),\ x_{10} = f_3(x_9, x_7),\ x_{11} = f_4(x_9, x_7)\ \}$.

Obviously, at this step of the modeling process, the function f_1, f_2, f_3 and f_4 cannot be entirely specified.

The behavioral model of Sapins' Dam is based on the definition of the set $C = \{C_i\}$ of observation classes where each $C_i = \{(x_i, \delta_{ij}\}$ a singleton (cf. Table 6.3).

Variable	**Definition Domain**	**Observation Classes**
x_7	$\Delta_{x_7} = \{\delta_{70}, \delta_{71}, \delta_{72}\}$	$C_{70} = \{(x_7, \delta_{70})\}$, $C_{71} = \{(x_7, \delta_{71})\}$, $C_{72} = \{(x_7, \delta_{72})\}$
x_8	$\Delta_{x_8} = \{\delta_{80}, \delta_{81}, \delta_{82}\}$	$C_{80} = \{(x_8, \delta_{80})\}$, $C_{81} = \{(x_8, \delta_{81})\}$, $C_{82} = \{(x_8, \delta_{82})\}$
x_9	$\Delta_{x_9} = \{\delta_{90}, \delta_{91}, \delta_{92}\}$	$C_{90} = \{(x_9, \delta_{90})\}$, $C_{91} = \{(x_9, \delta_{91})\}$, $C_{92} = \{(x_9, \delta_{92})\}$
x_{10}	$\Delta_{x_{10}} = \{\delta_{100}, \delta_{101}, \delta_{102}\}$	$C_{100} = \{(x_{10}, \delta_{100})\}$, $C_{101} = \{(x_{10}, \delta_{101})\}$, $C_{102} = \{(x_{10}, \delta_{102})\}$
x_{11}	$\Delta_{x_{11}} = \{\delta_{110}, \delta_{111}, \delta_{112}\}$	$C_{110} = \{(x11, \delta_{110})\}$, $C_{111} = \{(x_{11}, \delta_{111})\}$, $C_{112} = \{(x_{11}, \delta_{112})\}$

Table 6.3: Definition of The Observation Classes

Table 6.4 shows the change of the variables value during time for the described scenario.

Time	x_7	x_8	x_9	x_{10}	x_{11}
11/1978	–	–	δ_{91}	–	–
12/1978	–	δ_{81}	δ_{91}	–	–
04/1979	–	δ_{81}	δ_{92}	–	–
11/1980	δ_{71}	δ_{81}	δ_{92}	–	–
11/1981	δ_{70}	δ_{81}	δ_{92}	–	–
11/1981	δ_{70}	δ_{80}	δ_{92}	–	–
12/1981	δ_{71}	δ_{80}	δ_{92}	–	–
12/1981	δ_{71}	δ_{81}	δ_{92}	–	–
02/1983	δ_{70}	δ_{81}	δ_{92}	–	–
12/1985	δ_{70}	δ_{80}	δ_{92}	–	–
03/1986	δ_{70}	δ_{81}	δ_{92}	–	–
09/1988	δ_{70}	δ_{81}	δ_{92}	–	δ_{112}
10/1988	δ_{70}	δ_{81}	δ_{92}	δ_{102}	δ_{112}

Table 6.4: Variables Evolution During the Sapins' Sam Scenario

Given Table 6.3 and 6.4, the scenario can be written in term of occurrences of observation classes (cf. figure 6.7):

- $\omega = \{\ C_{91}(11/1978), C_{81}(12/1978), C_{92}(04/1979), C_{71}(11/1980), C_{70}(11/1981), C_{80}(11/1981), C_{71}(12/1981), C_{81}(12/1981), C_{70}(02/1983), C_{80}(12/1985), C_{81}(03/1986), C_{112}(09/1988), C_{102}(10/1988)\ \}$.

According to the *Spatial Discretization Principle* of the TOT, the first observation $C_{91}(11/1978)$ cannot be written by a safe program $\Theta(X, \Delta)$ because the time functions $x_7(t)$, $x_8(t)$, $x_9(t)$,

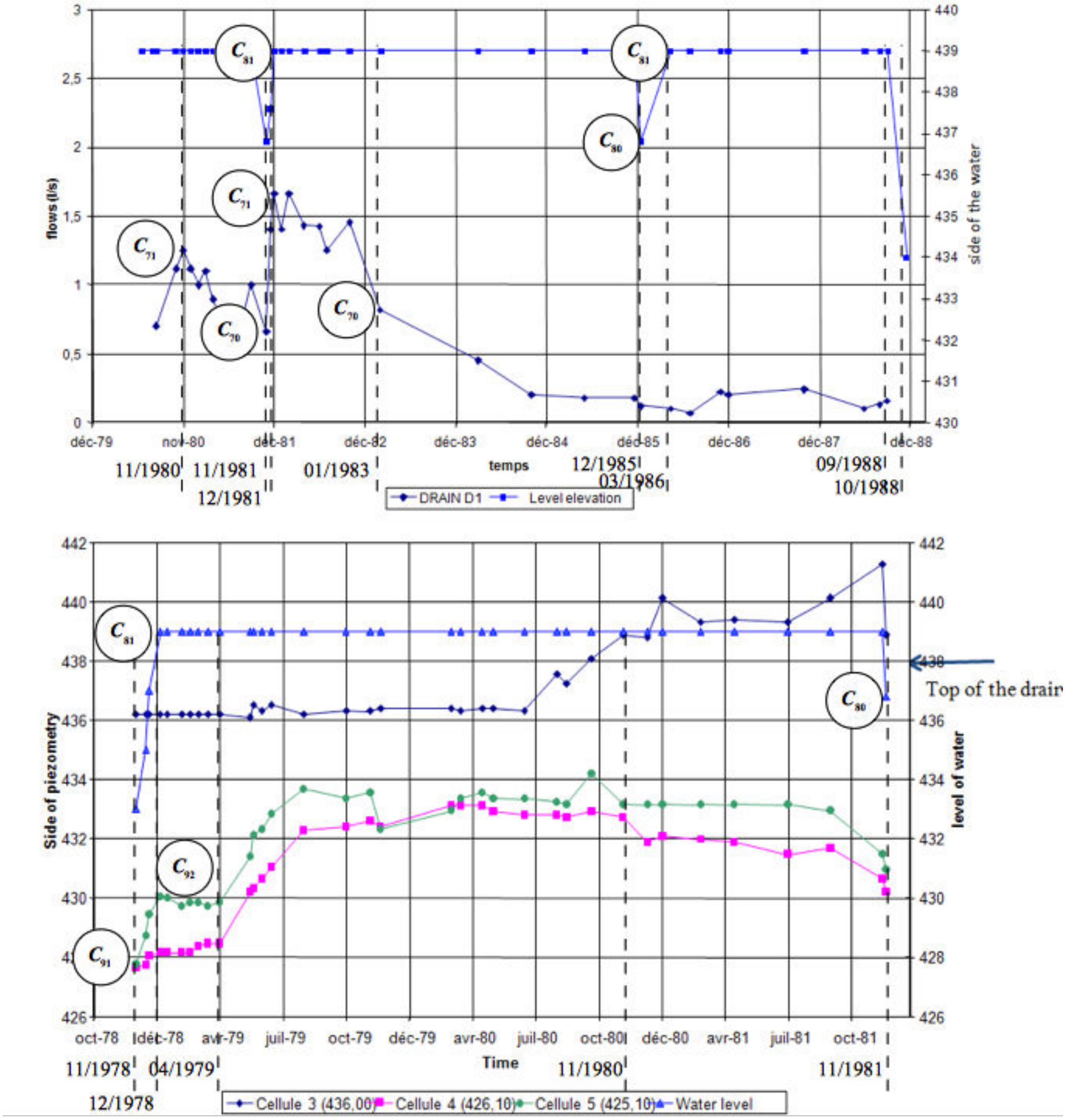


Figure 6.7: Sapins'Dam Degradation Sequence

$x_{10}(t)$ and $x_{11}(t)$ do not cross any threshold, the initial state is then unknown. At the beginning of the reservoir filling, the thresholds do not have the same values as after the reservoir is full.

So a first behavioral model can be depicted (cf. figure 6.8).

Formally, according to the definition **??** of TOM4D methodology, the behavioral model $BM(\Omega)$ is a structure $< S, C, R^s >$ where:

- $S = \{\ s_0,\ s_1,\ s_2,\ s_3,\ s_4,\ s_5,\ s_6,\ s_7,\ s_8,\ s_9,\ s_{10}, s_{11},\ s_{12},\ s_{13}\ \}$
- $C = \{\ C_{80} = \{(x_8, \delta_{80}\}),\ C_{81} = \{(x_8, \delta_{81})\},\ C_{82} = \{(x_8, \delta_{82})\},\ C_{90} = \{(x_9, \delta_{90})\},$ $C_{91} = \{(x_9, \delta_{91})\},\ C_{92} = \{(x_9, \delta_{92})\},\ C_{100} = \{(x_{10}, \delta_{100})\},\ C_{101} = \{(x_{10}, \delta_{101})\},$ $C_{102} = \{(x_{10}, \delta_{102})\},\ C_{110} = \{(x_{11}, \delta_{110})\},\ C_{111} = \{(x_{11}, \delta_{111})\},\ C_{112} = \{(x_{11}, \delta_{112})\}\ \}$
- $R^s = \{s_0 = \{r(C_\phi, C_{91})\}, s_1 = \{r(C_{91}, C_{81})\}, s_2 = \{r(C_{81}, C_{92})\}, s_3 = \{r(C_{92}, C_{71})\}, s_4 =$

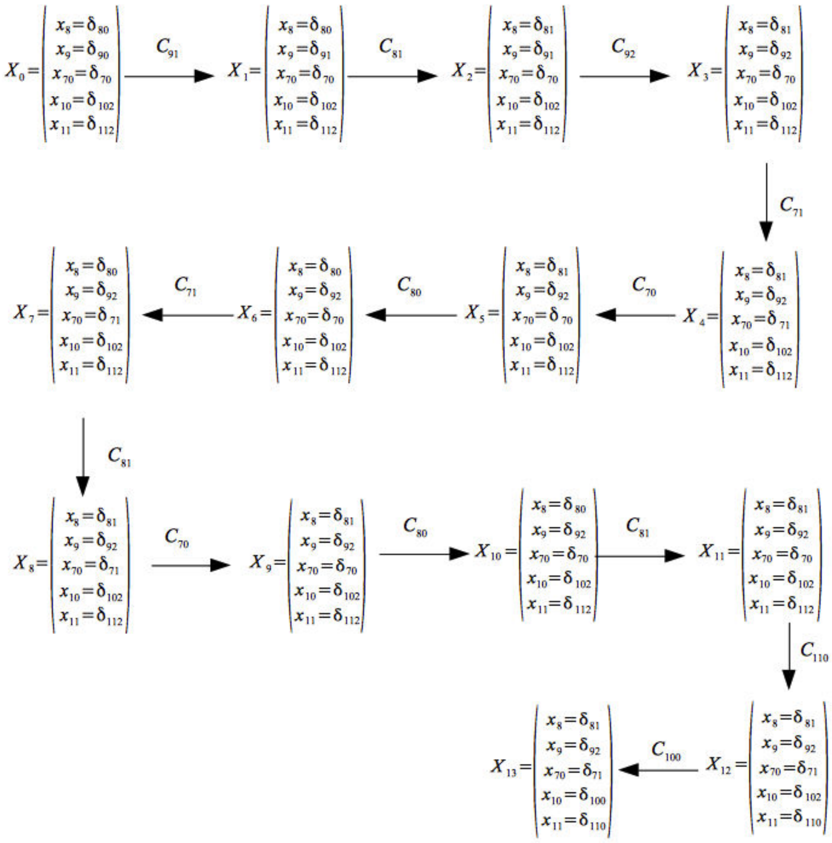


Figure 6.8: Behavior Model $BM(\Omega)$

$\{r(C_{71}, C_{70})\}$,
$s_5 = \{r(C_{70}, C_{80})\}, s_6 = \{r(C_{80}, C_{71})\}, s_7 = \{r(C_{71}, C_{81})\}, s_8 = \{r(C_{81}, C_{70})\}, s_9 = \{r(C_{70}, C_{80})\}$,
$s_{10} = \{r(C_{80}, C_{81})\}, s_{11} = \{r(C_{81}, C_{112})\}, s_{12} = \{r(C_{112}, C_{102})\}, \}$

Clearly, this model is only a part of the global behavioral model of the hydraulic process: potentially, the *discernible state space* contains $3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 = 243$ states. To build the complete behavioral model, $243 \cdot 242 = 58806$ relations between two discernible states must be examined. Moreover, analyzing Table 6.4, it can be seen that the functional model is inevitably incomplete. Indeed, at 12/1978 and 04/1979, $x_8 = \delta_{81}$, while $x_9 =$ respectively δ_{91} and δ_{92}. This leads us suppose that other variables are necessary to define the functioning of the dam, even if only five timed observations are available for the analysis of the dam. Consequently, a deeper analysis is then required: this is the role of the two next steps of the TOM4D modeling process.

6.4 Process definition

The first task of the Process Definition step consists in the definition of the Sapins' Dam under the form of a set $X(t)$ of time functions. The models made in the precedent step allow to immediately identify $x_7(t)$, $x_8(t)$, $x_9(t)$, $x_{10}(t)$ and $x_{11}(t)$ as time functions:

- x_7, x_{10} and x_{11} denote respectively the water output flow of the drain ($DrainFlow$) and a presence of water (*Leakage*, $WetArea$) on the downstream shoulder ($WaterDown$): they correspond to the Q variable of the hydraulic ToS. The physical dimension of x_7, x_{10} and x_{11} is then $m^3.s^{-1}$.

- x_8 denotes the water level of the reservoir ($ResLevel$): it corresponds to the V variable of the hydraulic ToS. The physical dimension of x_8 is m^3.

- x_9 denotes a pressure in the upstream shoulder ($PressureCell$). The physical dimension of x_9 is $kg/m.s^2$.

To define the Perception Model, we rely on the literature. The goal of an hydraulic dam is to generate a reservoir and to maintain the level of water considering an operating curve. The normal functioning must consider this goal along with safety aspects for people and assets located downstream. This implies that water that seeps into the upstream embankment (or foundation - not treated here) is correctly evacuated by a drainage system. Consequently, a normal functioning mode implies that there is no water in the downstream shoulder. This leads to the following definition of the Perception Model:

Definition 6.1 *The Perception Model $PM(X(t))$ of the hydraulic process $X(t)$ =\{ $x_7(t)$, $x_8(t)$, $x_9(t)$, $x_{10}(t)$, $x_{11}(t)$\} is the structure $< X, \Psi, R^q >$ where:*

- $X = \{x_7, x_8, x_9, x_{10}, x_{11}\}$,

- $\Psi = \{\Psi_{71}, \Psi_{72}, \Psi_{81}, \Psi_{82}, \Psi_{91}, \Psi_{92}, \Psi_{101}, \Psi_{111}\}$

- $R^q = R^{goal} \cup R^n \cup R^{ab}$.

    - $R^{goal} = \{\exists t_0, \exists t_i, t_i \geq t_o, \forall t \geq t_i, \Psi_{81} \leq x_8(t) < \Psi_{82}\}$
    - $R^n = \{$
      $\forall t \geq t_i, \Psi_{81} \leq x_8(t) < \Psi_{82} \wedge$
      $\forall t \geq t_i, \Psi_{71} \leq x_7(t) < \Psi_{72} \wedge$
      $\forall t \geq t_i, \Psi_{91} \leq x_9(t) < \Psi_{92}\} \wedge$
      $\forall t, x_{10}(t) < \Psi_{101} \wedge$
      $\forall t, x_{11}(t) < \Psi_{111}$
      $\}$
    - $R^{ab} = \{$
      $\forall t \geq t_i, x_8(t) < \Psi_{81} \vee x_8(t) \leq \Psi_{82} \vee$
      $\forall t \geq t_i, x_7(t) < \Psi_{71} \vee x_7(t) \leq \Psi_{72} \vee$
      $\forall t \geq t_i, x_9(t) < \Psi_{91} \vee x_9(t) \leq \Psi_{92} \vee$
      $\forall t, x_{10}(t) \geq \Psi_{101} \vee$
      $\forall t, x_{11}(t) \geq \Psi_{111}$
      $\}$

The perception model $PM(X(t))$ being defined, it is then possible to go in the last modeling step, the *Generic Modeling*.

6.5 Generic Modeling

Let us consider the hydraulic component of figure 6.9 that is made with one input variable $Q(t)$, the input flow, and two output variables $Q_s(t)$ (normal output flow) and $Q_f(t)$ (leaking output flow). In normal condition, $Q_f(t) = 0$ (i.e. there is no leaking).

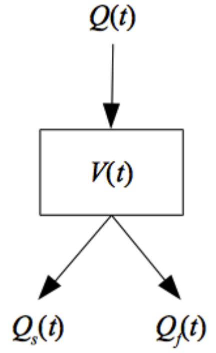


Figure 6.9: Hydraulic Component

The relations between the variables $Q(t)$, $V(t)$, $Q_s(t)$ and $Q_f(t)$ are defined by the hydraulic ToS (cf. figure 6.10).

6.5.1 Generic Interpretation with the ToS

By definition, the generalized flow $Q_v(t)$ corresponds to the internal volume of water inside the component:

$$Qv(t) = Q(t) - (Q_s(t) + Q_f(t)) \tag{6.1}$$

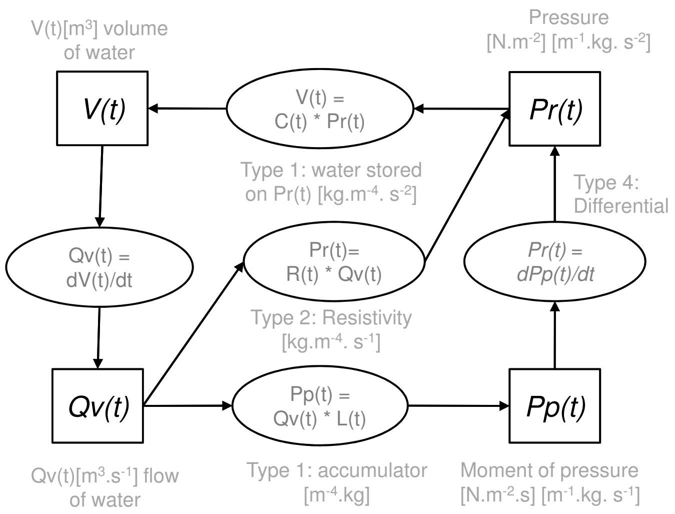

V(t)[m³] volume of water
Pressure [N.m⁻²] [m⁻¹.kg. s⁻²]
V(t)
V(t) = C(t) * Pr(t)
Pr(t)
Type 1: water stored on Pr(t) [kg.m⁻⁴. s⁻²]
Type 4: Differential
Qv(t) = dV(t)/dt
Pr(t)= R(t) * Qv(t)
Pr(t) = dPp(t)/dt
Type 2: Resistivity [kg.m⁻⁴. s⁻¹]
Qv(t)
Pp(t) = Qv(t) * L(t)
Pp(t)
Qv(t)[m³.s⁻¹] flow of water
Type 1: accumulator [m⁻⁴.kg]
Moment of pressure [N.m⁻².s] [m⁻¹.kg. s⁻¹]

Figure 6.10: HydraulicToS

When considering the components of an hydraulic dam, the momentum of pressure is not kept the experts' hydraulic dam. $Pr(t)$ is not observable for all components. The behavior of the hydraulic component of figure 6.9 is then given by the following equations:

$$Qv(t) = dV(t)/dt \tag{6.2}$$

$$Pr(t) = R(t) \cdot Qv(t) \tag{6.3}$$

$$V(t) = C(t) \cdot Pr(t) = C(t) \cdot R(t) \cdot Qv(t) \tag{6.4}$$

$$Qv(t) = C(t) \cdot d(R(t) \cdot d(Qv(t))/dt \tag{6.5}$$

As a consequence:

$$Q_s(t) + Q_f(t) = Q(t) - d(C(t) \cdot R(t) \cdot Qv(t))/dt \tag{6.6}$$

This equation means that the behavior of the dam components rely on the evolution of the product $C(t) \cdot R(t)$. This later is similar to the opposite of the permeability: the higher the product $C(t) \cdot R(t)$, the lower the permeability, and vice-versa.

6.5.2 Type of Components

Let us denote $\phi(t) = C(t) \cdot R(t)$. The behavior of the dam component differs strongly according to three types of value $\phi(t)$ can take over time:

- $\phi(t) \to 0$: In that case, the dam component behaves as a *pipe*. A pipe is conceptually characterized by the fact that there is no storage (i.e. $V(t) = 0$) : its input flow equals its output flow with some delay (equation 6.7) .

$$Q_s(t + \Delta T) = Q(t) \tag{6.7}$$

- $\phi(t) \to \infty$: In that case, the hydraulic component behaves as a *tank*. A tank is characterized by its capacity $C(t) = C$ corresponding to the maximum quantity of water that be contained in it. Under normal condition, a tank has no outflow (i.e. $Q_s(t) = 0$ and $Q_v(t) = Q(t)$, equation 6.8).

$$Q(t) = dV(t)/dt \tag{6.8}$$

- $\phi(t) \to \phi_0$: In that case, the hydraulic component behaves as a *low pass filter*. In that case, the component is characterized by the value ϕ_0. In normal conditions ϕ_0 is a constant. Its role is to control the output flow $Qs(t)$ according to the current value of $C(t) \cdot R(t)$ product .

$$Q_s(t) = Q(t) - \phi_0 \cdot Qv(t)/dt \tag{6.9}$$

As a consequence, the equation 6.6 can be simplified as following where $\phi(t)$ is a variable that can take three *constant* values, 0, ϕ_0 and ∞:

$$Q_s(t) + Q_f(t) = Q(t) - \phi(t) \cdot dQv(t)/dt \tag{6.10}$$

Technically speaking, consider that the variable $\phi(t)$ can take three types of constant value over time simply means that the evolution of $\phi(t)$ over time is considered to be very slow according to the dynamic of the evolution of the other variables $Q_s(t)$, $Q_f(t)$, $Q(t)$ and $Qv(t)$. $\phi(t)$ is then considered as a *structural and non observable variable* of the component: the equation 6.10 aims at linking the values of $V(t)$, $Qs(t)$ and $Qf(t)$ related to the modifications of $\phi(t)$. In other words, $\phi(t)$ is an internal variable that governs the behavior of the dam components according to three types of behaviors:

- $\phi(t) = 0$: the component behaves as a *Pipe*.
- $\phi(t) = \phi_0$: the component behaves as a *Low Pass Filter*.
- $\phi(t) = \infty$: the component behaves as a *Tank*.

By construction, the evolution of the values of $\phi(t)$ characterizes that of the component permeability: this latter should stay constant over time. This means that in normal operation, $\phi(t)$ of a dam component is unique, whatever its value 0, ϕ_0 or ∞. But ageing, poor design or wrong building conditions can lead to modifications of the permeability of a component, that is to say $\phi(t)$ can take one (or two) abnormal value(s). When considering the components of an hydraulic dam, various directions are then possible for the values of $\phi(t)$:

- $\phi_0(t) = 0 \to \phi(t + \Delta T_1) = \phi_0 \to \phi(t + \Delta T_1 + \Delta T_2) = \infty$: a *pipe* can become a *low pass filter*, and next a *tank*.
- $\phi(t) = \phi_0 \to \phi(t + \Delta T_3) = \infty$: a *low pass filter* can become a *pipe*.

Conversely, a *tank* (i.e. $\phi(t) = \infty$) can never become a *low pass filter* (or a *pipe*) without an intervention on the dam (i.e. the reparation of a component). These properties of $\phi(t)$ will have a strong impact on the TOM4D behavioral models, but the important point here is that $\phi(t)$ being a *structural variable*, the only way to measure (and so to observe) its value is through the demolition of the dam. In other words, a physical (or concrete) component can play the role of two or even three types of component. All happens as if the original component, say a pipe for example, would be replaced by another one, say a low pass filter. As a consequence, the only way to determine the value of $\phi(t)$ without demolishing the component is to detect its current behavior: for example, if the component behaves like a low pass filter when it should behave like a pipe, then this means that $\phi(t)$ does not get the value the design assigned to it (i.e. $\phi_0(t) = 0$) but something happens so that $\phi(t) = \phi_0$. In other words, and this is the point of view that is retained in this document, the values of $\phi(t)$ constitute the *context* in which a component operates, and the role of the diagnosis is then to analyze the current behavior of a component in order to determine the *current operating context*.

So, the equation 6.10 will be used in order to characterize the different components of the Sapins hydraulic dam as *Tank*, *Pipe* and *Low Pass Filter* in normal operations:

1. The Vertical Drain c_1 is a *pipe*: all the water coming from the Upstream Shoulder c_3 goes through the Vertical Drain c_1.

2. The Reservoir c_2 is a *pipe* that acts as a *generator* of a water flow: there is no resistance to the external incoming water. The different values of *ResLevel* (i.e. x_8) come from a more or less important supply of water.

3. The Upstream Shoulder c_3 is a *low pass filter*: only a few amount of water goes from the Reservoir c_2 to the Upstream Shoulder c_3. This latter opposes the ingress of water but not totally because its materials can not be completly watertight;

4. The Downstream Shoulder c_4 is a *tank*. The term *Leakage*(δ_{100}) and $WetArea$ (δ_{110}) associated to the Downstream Shoulder c_4 indicate abnormal operating conditions.

5. The Horizontal Drain c_6 is a *pipe*: all the water coming from the Foundation goes through the Horizontal Drain c_6;

In nomal condition, the value of the variable $Q_f(t)$ is null for each type of component and when something abnormal occurs in a component, the value of $Q_f(t)$ is positive.

6.5.3 Dynamic Process Network Model for Sapin's Dam

Let us denote (cf. Table 6.5):

- ϕ_{21} the structural variable of the *Vertical Drain* c_1. ϕ_{21} is defined over $\Delta_{\phi 21}$: the values δ_{210}, δ_{211} and δ_{212} correspond respectively to $\phi_{c_1}(t) \rightarrow 0$ (Low resistance to water, c_1 behaves like a pipe), $\phi_{c_1}(t) \rightarrow \phi_0$ (Medium resistance to water, c_1 behaves like a low pass filter) and $\phi_{c_1}(t) \rightarrow \infty$ (High resistance to water, c_1 behaves like a tank).
- ϕ_{23} the internal variable of the *Upstream Shoulder* c_3. ϕ_{23} is defined over $\Delta_{\phi 23}$: the values δ_{230} and δ_{231} correspond respectively to $\phi_{c_3}(t) \rightarrow 0$ (Low resistance to water, c_3 behaves like a pipe) and $\phi_{c_3}(t) \rightarrow \phi_0$ (Medium resistance to water, c_3 behaves like a low pass filter).

In other words, the values of ϕ_{21} represent the fact the *Vertical Drain* c_1 that functions under normal conditions as a *pipe* (Low resistance, $\phi_{21} = \delta_{210}$) can become a *low pass filter* (Medium resistance, $\phi_{21} = \delta_{211}$) and then a *tank* (High resistance, $\phi_{21} = \delta_{212}$). Conversely, we consider that the *Upstream Shoulder* that has Medium resistance under normal conditions (i.e. $\phi_{23} = \delta_{231}$) can become a *pipe* with Low resistance (i.e. $\phi_{23} = \delta_{230}$).

Var.	**Definition Domain**	**Constant Name**
ϕ_{21}	$\Delta_{\phi_{21}} = \{\delta_{210}, \delta_{211}, \delta_{212}\}$	Low resistance, Medium resistance, High resistance
ϕ_{23} -	$\Delta_{\phi_{23}} = \{\delta_{230}, \delta_{231}\}$	Low resistance, Medium resistance

Table 6.5: Definition of the Variables ϕ_{21} and ϕ_{23}

Var.	**Definition Domain**	**Respective Abstract Ranges**
ϕ_{21}	$\Delta_{\phi_{21}} = \{\delta_{210}, \delta_{211}, \delta_{212}\}$	$]-\infty, \Psi_{211}[$, $[\Psi_{211}, \Psi_{212}[$, $[\Psi_{212}, +\infty[$
ϕ_{23}	$\Delta_{\phi_{23}} = \{\delta_{230}, \delta_{231}\}$	$]-\infty, \Psi_{231}[$, $[\Psi_{231}, +\infty[$

Table 6.6: Definition of The Abstract Thresholds for ϕ_{21} and ϕ_{23}

Using these internal variables ϕ_{21} and ϕ_{23}, the functional model can be updated (cf. Figure 6.11).

Formaly, the Generic Functional Model $FM(X(t))$ of the dam is given in the definition 6.2.

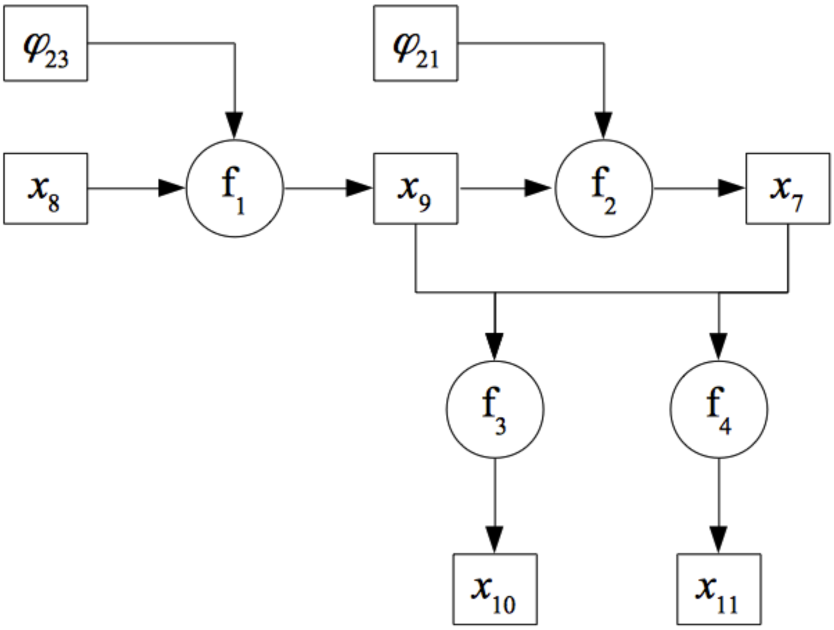


Figure 6.11: Generic Functional Model $FM(X(t))$

Definition 6.2 *The Functional Model $FM(X(t))$ of the hydraulic process $X(t)$ is the structure $< \Delta, F, R^f >$ where,*

- $\Delta = \Delta_{x_7} \cup \Delta_{x_8} \cup \Delta_{x_9} \cup \Delta_{x_{10}} \cup \Delta_{x_{11}} \cup \Delta_{\phi_{21}} \cup \Delta_{\phi_{23}}$.
- $F = \{$
  $f_1 : \Delta^{x_8} \times \Delta^{\phi_{23}} \times \Delta^{x_9}$,
  $f_2 : \Delta^{x_9} \times \times \Delta^{\phi_{21}} \times \Delta^{x_7}$,
  $f_3 : \Delta^{x_9} \times \Delta^{x_7} \times \Delta^{x_{10}}$,
  $f_4 : \Delta^{x_9} \times \Delta^{x_7} \times \Delta^{x_{11}}$
  $\}$.
- $R^f = \{$
  $x_9 = f_1(x_8, \phi_{23})$,
  $x_7 = f_2(x_9, \phi_{21})$,
  $x_{10} = f_3(x_9, x_7)$,
  $x_{11} = f_4(x_9, x_7)$
  $\}$.

The functions f_1, f_2, f_3 and f_4 are specified in the tables 6.7, 6.8, 6.9.

This functional model leads naturally to define the hydraulic process $X(t)$ as a network of dynamic processes $X(t) = X_{11}(t) \cup X_{12}(t) \cup X_{13}(t)$ where:

- $X_{11}(t)$ implements the function f_1: $X_{11}(t) = \{x_8(t), x_9(t), \phi_{23}(t)\}$.
- $X_{12}(t)$ implements the function f_2: $X_{12}(t) = \{x_9(t), x_7(t), \phi_{21}(t)\}$.
- $X_{13}(t)$ implements the functions f_3 and f_4: $X_{13}(t) = \{x_7(t), x_9(t), x_{10}(t), x_{11}(t)\}$.

$f_1 : \Delta_{x_8} \times \Delta_{\phi_{23}} \rightarrow \Delta_{x_9}$ $x_9 = f_1(x_8, \phi_{23})$			
x_8	ϕ_{23}	f_1	*Interpretation*
δ_{80}	δ_{231}	δ_{90}	*Low ResLevel, Medium UpShoulder Resistance* ⇒ *Low PressureCell*
δ_{81}	δ_{231}	δ_{91}	*Medium ResLevel, Medium UpShoulder Resistance* ⇒ *Medium PressureCell*
δ_{82}	$-$	δ_{92}	*High ResLevel* ⇒ *High PressureCell*
δ_{80}	δ_{230}	δ_{91}	*Low ResLevel, Low UpShoulder Resistance* ⇒ *Medium PressureCell*
δ_{81}	δ_{230}	δ_{92}	*Medium ResLevel, Low UpShoulder Resistance* ⇒ *High PressureCell*

Table 6.7: Definition of the Generic Function f_1

$f_2 : \Delta_{x_9} \times \Delta_{\phi_{21}} \rightarrow \Delta_{x_7}$ $x_7 = f_2(x_9, \phi_{21})$			
x_9	ϕ_{21}	x_7	*Interpretation*
δ_{90}	δ_{210}	δ_{70}	*Low PressureCell, Low Drain Resistance* ⇒ *Low Drain Flow*
δ_{91}	δ_{210}	δ_{71}	*Medium PressureCell, Low Drain Resistance* ⇒ *Medium Drain Flow*
δ_{92}	δ_{210}	δ_{72}	*High PressureCell, Low Drain Resistance* ⇒ *High Drain Flow*
δ_{90}	δ_{211}	δ_{70}	*Low PressureCell, Medium Drain Resistance* ⇒ *Low Drain Flow*
δ_{91}	δ_{211}	δ_{70}	*Medium PressureCell, Medium Drain Resistance* ⇒ *Low Drain Flow*
δ_{92}	δ_{211}	δ_{71}	*High PressureCell, Medium Drain Resistance* ⇒ *Medium Drain Flow*
$-$	δ_{212}	δ_{70}	*High Drain Resistance* ⇒ *Low Drain Flow*

Table 6.8: Definition of the Generic Function f_2

$f_3 : \Delta_{x_9} \times \Delta_{x_7} \rightarrow \Delta_{x_{10}}, f_4 : \Delta_{x_9} \times \Delta_{x_7} \rightarrow \Delta_{x_{11}}$ $f_3(x_9, x_7), f_4(x_9, x_7)$				
x_9	x_7	f_3	f_4	*Interpretation*
δ_{90}	δ_{70}	δ_{100}	δ_{110}	*Low PressureCell, Low DrainFlow* ⇒ *AbsLeak, AbsWA*
δ_{91}	δ_{71}	δ_{100}	δ_{110}	*Medium PressureCell, Medium DrainFlow* ⇒ *AbsLeak, AbsWA*
δ_{92}	δ_{72}	δ_{100}	δ_{110}	*High PressureCell, High DrainFlow* ⇒ *AbsLeak, AbsWA*
δ_{91}	δ_{70}	δ_{101}	δ_{111}	*Medium PressureCell, Low DrainFlow* ⇒ *SuspLeak, SuspWA*
δ_{92}	δ_{71}	δ_{101}	δ_{111}	*High PressureCell, Medium DrainFlow* ⇒ *SuspLeak, SuspWA*
δ_{92}	δ_{70}	δ_{102}	δ_{112}	*High PressureCell, Low DrainFlow* ⇒ *PresLeak, PresWA*

Table 6.9: Definition of the Generic Functions f_3 and f_4

In the functional model of figure 6.11, the sensors c_7 $(DrainFlow)$, c_8 $(ResLevel)$, c_9 $(PressureCell)$, c_{10} $(VisualObs)$ are considered as always available: they can then been merged in their respective components c_1, c_2, c_3 and c_4. Moreover, the components c_2 (*Reservoir*) having no sensor, it can be considered as merged with *Upstream Shoulder* c_3. This leads to a simplified the generic structural model graphically represented in figure 6.12.

Definition 6.3 *The Generic Structural Model* $SM(X(t))$ *is a structure* $< COMPS, R^p, R^x >$ *where:*

- $COMPS = \{c_{11}, c_{12}, c_{13}\}$
- $R^p = \{out_1(c_{11}) = in_1(c_{12}), out_2(c_{11}) = in_1(c_{13}), out_1(c_{12}) = in_2(c_{13})\}$
- $R^x = \{x_8 = in_1(c_{11}), \phi_{23} = in_2(c_{11}), x_9 = out_3(c_{11}), x_7 = out_2(c_{12}), \phi_{21} = in_2(c_{12}), x_{10} = out_1(c_{13}), x_{11} = out_2(c_{13})\}$

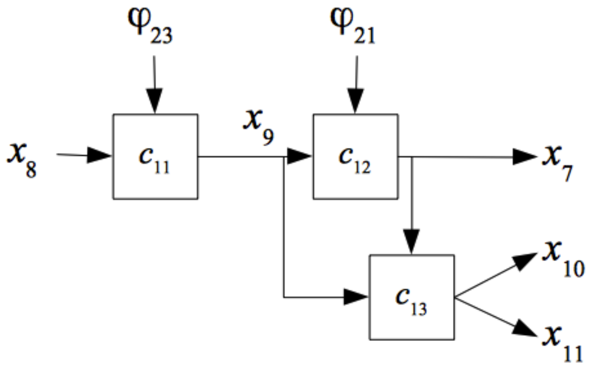


Figure 6.12: Generic Structural Model $SM(X(t))$

The interest of the network structure is the simplification of the building of the generic behavioral models of $X_{11}(t)$, $X_{12}(t)$ and $X_{13}(t)$. The delayed effects due to the water transfer into the different elements that are not represented in the functions f_1, f_2, f_3 and f_4 will be latter considered in the behavioral model. For instance, under normal operating conditions, an increase of the water elevation in the reservoir c_2 will result in an increase of the piezometry in the Upstream Shoulder c_3 but this change is not instantaneous. In a same way, the interactions between components will be considered in the behavioral models: in case of an increase of the Vertical Drain resistance R_{c_1}, i.e. $\phi_{21} = \delta_{211}$ (Medium resistance) or δ_{212} (High resistance), the piezometry will be higher than the piezometry observed if the drain resistance is low.

6.5.4 Step 3.1: Generic Modeling of $X_{11}(t)$

The Perception Model of the process $X_{11}(t)$ is given in the Definition 6.4:

Definition 6.4 *The Perception Model $PM(X_{11}(t))$ of the process $X_{11}(t) = \{x_8(t), x_9(t), x_7(t)\}$ is the structure $< X_{11}, \Psi_{11}, R^{q_{11}} >$ where:*

- $X_{11} = \{x_8(t), x_9(t), \phi_{23}(t)\}$,
- $\Psi = \{\Psi_{81}, \Psi_{82}, \Psi_{91}, \Psi_{92}, \Psi_{231}\}$
- $R^{q_{11}} = R^{goal_{11}} \cup R^{n_{11}} \cup R^{ab_{11}}$.
  - $R^{goal_{11}} = \{\exists t_0, \exists t_i, t_i \geq t_0,$
    $\forall t \geq t_i, \Psi_{81} \leq x_8(t) < \Psi_{82}$
    $\}$
  - $R^{n_{11}} = \{\forall t \geq t_i,$
    $\Psi_{81} \leq x_8(t) < \Psi_{82} \wedge$
    $\Psi_{91} \leq x_9(t) < \Psi_{92} \wedge$
    $\}$
  - $R^{ab_{11}} = \{\forall t \geq t_i,$
    $x_8(t) < \Psi_{81} \vee x_8(t) \geq \Psi_{82} \vee$
    $x_9(t) < \Psi_{91} \vee x_9(t) \geq \Psi_{92} \vee$
    $\}$

The structural model of $X_{11}(t)$ is given in the definition 6.5 and is represented in the figure 6.13.

Definition 6.5 *The structural model $SM(X_{11}(t))$ is the structure $< COMPS_{11}, R^{p_{11}}, R^{x_{11}} >$ where:*

- $COMPS_{11} = \{c_2, c_3, c_8, c_9\}$
- $R^{p_{11}} = \{out(c_2) = in(c_3)\}$
- $R^{x_{11}} = \{x_8 = out(c_8), x_9 = out(c_9)\}$

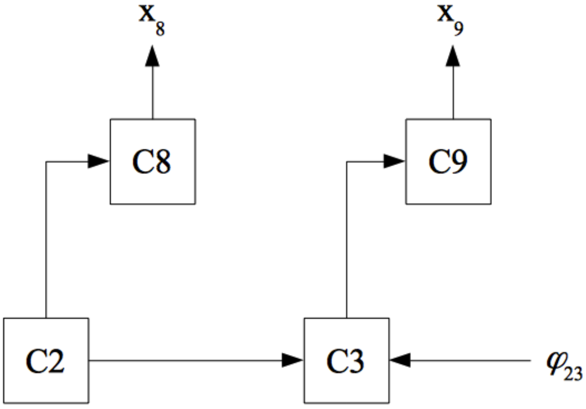


Figure 6.13: Generic Structural Model $SM(X_{11}(t))$

The Generic Functional Model $FM(X_{11}(t))$ of $X_{11}(t)$ is given in the definition 6.6 (cf. figure 6.14).

Definition 6.6 *The Functional Model $FM(X(t))$ of the hydraulic process $X(t)$ is the structure $< \Delta_{11}, F_{11}, R^{f_{11}} >$ where,*

- $\Delta_{11} = \Delta_{x_8} \cup \Delta_{\phi_{23}} \cup \Delta_{x_9}$.
- $F_{11} = \{f_1 : \Delta^{x_8} \times \Delta^{\phi_{23}} \times \Delta^{x_9}\}$.
- $R^f = \{x_9 = f_1(x_8, \phi_{23})\}$.

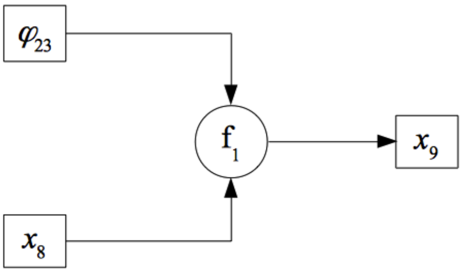


Figure 6.14: Generic Functional Model $FM(X_{11}(t))$

The relations between the values of the variables defined in the functional model of f_1 (6.7) allow to define the possible discernible states of $X_{11}(t)$. In the model below (cf. figure 6.15), it is important to note that *only the physically possible discernible states have been considered.* This means in particular that the following hypothesis has been used to model the behavior of $X_{11}(t)$: the water in the *Usptream Shoulder* is supposed to totally stem from the *Reservoir.* This hypothesis simplifies the behavioral model because only 13 discernible states have to be considered instead of the $3.3.3 = 27$ potential discernible states defined by $X_{11}(t)$. The behavioral model is formally given below:

Definition 6.7 *The behavior model $BM(X_{11}(t))$ of the dynamic process $X_{11}(t) = \{x_8(t), x_9(t), \phi_{23}\}$ is the structure $< S_{11}, C_{11}, R^{s_{11}} >$ where:*

- *The set* $R^{s_{11}} = \{$
  $s_{110} = \{\ r_{1101}(C_{80}, C_{81}),\ r_{1102}(C_{92}, C_{81})\ \}$
  $s_{111} = \{\ r_{1111}(C_{80}, C_{81}),\ r_{121}(C_{80}, C_{92})\ \}$
  $s_{112} = \{\ r_{1121}(C_{91}, C_{90}),\ r_{1122}(C_{80}, C_{81}),\ r_{1123}(C_{91}, C_{81}),\ r_{1124}(C_{80}, C_{90})\ \}$
  $s_{113} = \{\ r_{1131}(C_{90}, C_{81}),\ r_{1132}(C_{90}, C_{91})\ \}$
  $s_{114} = \{\ r_{1141}(C_{81}, C_{82}),\ r_{1142}(C_{81}, C_{91}),\ r_{1143}(C_{92}, C_{91}),\ r_{1144}(C_{92}, C_{82})\ \}$
  $s_{115} = \{\ r_{1151}(C_{81}, C_{80}),\ r_{1152}(C_{81}, C_{82}),\ r_{1153}(C_{92}, C_{80}),\ r_{1154}(C_{92}, C_{82})\ \}$
  $s_{116} = \{\ r_{1161}(C_{81}, C_{80}),\ r_{1162}(C_{81}, C_{82}),\ r_{1163}(C_{81}, C_{90}),\ r_{1164}(C_{81}, C_{92}),\ r_{1165}(C_{91}, C_{80}),$ $r_{1166}(C_{91}, C_{82}),\ r_{1167}(C_{91}, C_{90}),\ r_{1168}(C_{91}, C_{92})\ \}$
  $s_{117} = \{\ r_{1171}(C_{81}, C_{91}),\ r_{1172}(C_{90}, C_{91})\ \}$
  $s_{118} = \{\ r_{1181}(C_{92}, C_{81}),\ r_{1182}(C_{82}, C_{81})\ \}$
  $s_{119} = \{\ r_{1191}(C_{82}, C_{81}),\ r_{1192}(C_{92}, C_{81})\ \}$
  $s_{120} = \{\ r_{1201}(C_{82}, C_{81}),\ r_{1202}(C_{82}, C_{92})\ \}$
  $s_{121} = \{\ r_{1211}(C_{81}, C_{80}),\ r_{1212}(C_{81}, C_{92}),\ r_{1213}(C_{81}, C_{82})\ \}$
  $s_{122} = \{\ r_{1221}(C_{82}, C_{81}), r_{1222}(C_{82}, C_{92})\ \}$
  $\}$
  of 39 sequential binary relations.

- *The set* $C_{11} = \{C_{80} = \{(x_8, \delta_{80})\}, C_{81} = \{(x_8, \delta_{81})\}, C_{82} = \{(x_8, \delta_{82})\}, C_{90} = \{(x_9, \delta_{90})\}, C_{91} = \{(x_9, \delta_{91})\}, C_{92} = \{(x_9, \delta_{92})\}\}$ *of 6 observation classes.*

- *The set* $R^{s_{11}} = \{$
  $s_{110} = \gamma(s_{111}, C_{92}),\ s_{110} = \gamma(s_{115}, C_{80}),$
  $s_{111} = \gamma(s_{112}, C_{80}),$
  $s_{112} = \gamma(s_{113}, C_{91}),\ s_{112} = \gamma(s_{116}, C_{80}),$
  $s_{113} = \gamma(s_{112}, C_{90}),$
  $s_{121} = \gamma(s_{111}, C_{81}),\ s_{121} = \gamma(s_{122}, C_{81}),$
  $s_{116} = \gamma(s_{112}, C_{81}),\ s_{116} = \gamma(s_{117}, C_{91}),\ s_{116} = \gamma(s_{120}, C_{81}),\ s_{116} = \gamma(s_{114}, C_{91}),$
  $s_{117} = \gamma(s_{113}, C_{81}),\ s_{117} = \gamma(s_{116}, C_{90}),$
  $s_{122} = \gamma(s_{121}, C_{82}),$

$s_{120} = \gamma(s_{116}, C_{82}),$
$s_{119} = \gamma(s_{122}, C_{92}),\ s_{119} = \gamma(s_{115}, C_{82}),$
$s_{118} = \gamma(s_{120}, C_{92}),\ s_{118} = \gamma(s_{114}, C_{82}),$
$s_{115} = \gamma(s_{119}, C_{81}),\ s_{115} = \gamma(s_{121}, C_{92}),\ s_{115} = \gamma(s_{110}, C_{81}),$
$s_{114} = \gamma(s_{118}, C_{81}),\ s_{114} = \gamma(s_{116}, C_{92})$
$\}$ *of 25 predicates.*

The figure 6.15 provides a graphical representation of $BM(X_{11}(t))$.

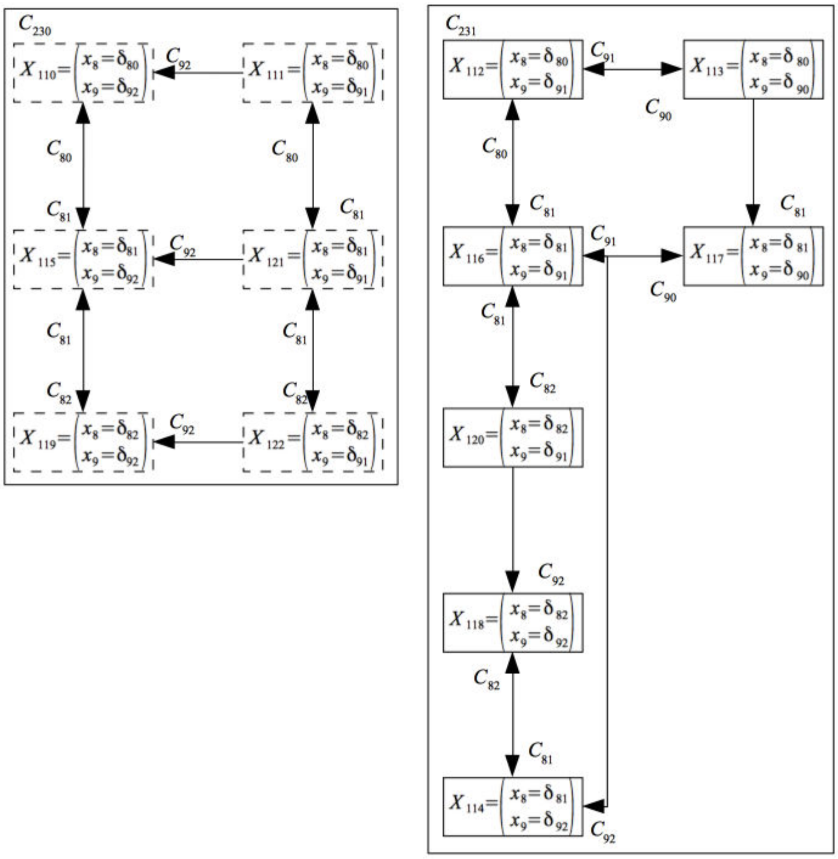

C_{230}
$X_{110} = \begin{pmatrix} x_8 = \delta_{80} \\ x_9 = \delta_{92} \end{pmatrix}$
C_{92}
$X_{111} = \begin{pmatrix} x_8 = \delta_{80} \\ x_9 = \delta_{91} \end{pmatrix}$
C_{80}
C_{80}
C_{81}
C_{81}
$X_{115} = \begin{pmatrix} x_8 = \delta_{81} \\ x_9 = \delta_{92} \end{pmatrix}$
C_{92}
$X_{121} = \begin{pmatrix} x_8 = \delta_{81} \\ x_9 = \delta_{91} \end{pmatrix}$
C_{81}
C_{81}
C_{82}
C_{82}
$X_{119} = \begin{pmatrix} x_8 = \delta_{82} \\ x_9 = \delta_{92} \end{pmatrix}$
C_{92}
$X_{122} = \begin{pmatrix} x_8 = \delta_{82} \\ x_9 = \delta_{91} \end{pmatrix}$
C_{231}
$X_{112} = \begin{pmatrix} x_8 = \delta_{80} \\ x_9 = \delta_{91} \end{pmatrix}$
C_{91}
$X_{113} = \begin{pmatrix} x_8 = \delta_{80} \\ x_9 = \delta_{90} \end{pmatrix}$
C_{90}
C_{80}
C_{81}
C_{81}
$X_{116} = \begin{pmatrix} x_8 = \delta_{81} \\ x_9 = \delta_{91} \end{pmatrix}$
C_{91}
$X_{117} = \begin{pmatrix} x_8 = \delta_{81} \\ x_9 = \delta_{90} \end{pmatrix}$
C_{90}
C_{81}
C_{82}
$X_{120} = \begin{pmatrix} x_8 = \delta_{82} \\ x_9 = \delta_{91} \end{pmatrix}$
C_{92}
$X_{118} = \begin{pmatrix} x_8 = \delta_{82} \\ x_9 = \delta_{92} \end{pmatrix}$
C_{82}
C_{81}
$X_{114} = \begin{pmatrix} x_8 = \delta_{81} \\ x_9 = \delta_{92} \end{pmatrix}$
C_{92}

Figure 6.15: Generic Behavioral Model $BM(X_{11}(t))$

In this Figure, each state s_k has been replaced with its corresponding *discernible state vector* X_k defined on $\{\Delta^{x_8} \times \Delta^{x_9} \times \Delta^{x_{23}}\}$ to facilitate the reading of the model. For example, the state $X_{111} = \{x_8 = \delta_{80}, x_{23} = \delta_{230}, x_9 = \delta_{91}\}$ represents the fact that there is few water in the *Reservoir*, the *Resistivity* of the *UpShoulder* is Low and the Piezometry in the *UpShoulder* is *Medium*. This behavioral model is composed of two behavioral sub-models: one characterizing the upstream shoulder behaving as a pipe (ϕ_{230}) and the other characterizing the upstream shoulder behaving as a low pass filter (ϕ_{231}) which is the normal behavior for this component.

It is to note that the model is non *symmetric* according to the values of x_{23} : once the resistance of the Upstream Shoulder gets the value *Low*, it is impossible to come back to a value *Medium* except in case of corrective actions but in that case, we consider that the dam is a new one.

6.5.5 Step 3.2: Generic Modeling of $X_{12}(t)$

The Perception Model of the process $X_{12}(t)$ is given in the Definition 6.8:

Definition 6.8 *The Perception Model $PM(X_{12}(t))$ of the process $X_{12}(t) = \{x_9(t), \phi_{21}, x_7(t)\}$ is the structure $< X_{12}, \Psi_{12}, R^{q_{12}} >$ where:*

- $X_{12} = \{x_9(t), \phi_{21}, x_7(t)\}$,
- $\Psi = \{\Psi_{91}, \Psi_{92}, \Psi_{71}, \Psi_{72}, \Psi_{211}, \Psi_{212}\}$
- $R^{q_{12}} = R^{goal_{12}} \cup R^{n_{12}} \cup R^{ab_{12}}$.
  - $R^{goal_{12}} = \{\Phi\}$ *(no goal can be defined: $X_{12}(t)$ must work normally)*
  - $R^{n_{12}} = \{\forall t \geq t_i,$
    $\Psi_{91} \leq x_9(t) < \Psi_{92} \wedge$
    $\Psi_{71} \leq x_7(t) < \Psi_{72}$
    $\}$
  - $R^{ab_{12}} = \{\forall t \geq t_i,$
    $x_7(t) < \Psi_{71} \vee x_7(t) \geq \Psi_{72} \vee$
    $x_9(t) < \Psi_{91} \vee x_9(t) \geq \Psi_{92}$
    $\}$

The structural model of $X_{12}(t)$ is given in the definition 6.9 and is represented in the figure 6.16.

Definition 6.9 *The structural model $SM(X_{12}(t))$ is the structure $< COMPS_{12}, R^{p_{12}}, R^{x_{12}} >$ where:*

- $COMPS_{12} = \{c_3, c_1, c_9, c_7\}$
- $R^{p_{12}} = \{out(c_3) = in(c_1)\}$
- $R^{x_{12}} = \{x_9 = out(c_9), x_7 = out(c_7)\}$

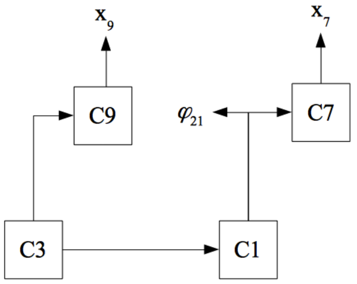


Figure 6.16: Generic Structural Model $SM(X_{12}(t))$

The Generic Functional Model $FM(X_{12}(t))$ of $X_{12}(t)$ is given in the definition 6.10 (cf. figure 6.17).

Definition 6.10 *The Functional Model $FM(X(t))$ of the hydraulic process $X(t)$ is the structure $< \Delta_{12}, F_{12}, R^{f_{12}} >$ where,*

- $\Delta_{12} = \Delta_{x_9} \cup \Delta_{x_7} \cup \Delta_{\phi_{21}}$.
- $F_{12} = \{f_2 : \Delta^{x_9} \times \Delta^{\phi_{21}} \times \Delta^{x_7}\}$.
- $R^f = \{x_7 = f_2(x_9, \phi_{21})\}$.

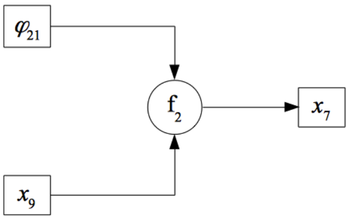


Figure 6.17: Generic Functional Model $FM(X_{12}(t))$

To define the possible states concerning $X_{12}(t)$, we rely on the values defined in the functional model (6.7). The behavioral model is given below:

Definition 6.11 *The behavior model $BM(X_{12}(t))$ of the dynamic process $X_{12}(t) = \{x_9(t), \phi_{21}(t), x_7(t)\}$ is the structure $< S_{12}, C_{12}, R^{s_{12}} >$ where:*

- *The set $R^{s_{12}}$ =\{*

  $s_{123} = \{\ r_{1231}(C_{70}, C_{91}), r_{1232}(C_{90}, C_{91})\ \}$

  $s_{124} = \{\ r_{1241}(C_{70}, C_{71}), r_{1242}(C_{70}, C_{90}), r_{1243}(C_{91}, C_{90}), r_{1244}(C_{91}, C_{71})\ \}$

  $s_{125} = \{\ r_{1251}(C_{71}, C_{70}), r_{1252}(C_{71}, C_{92}), r_{1253}(C_{71}, C_{90}), r_{1254}(C_{91}, C_{92}), r_{1255}(C_{91}, C_{90}), r_{1256}(C_{91}, C_{70})$
  $\}$

  $s_{126} = \{\ r_{1261}(C_{92}, C_{91}), r_{1262}(C_{92}, C_{72})\ \}$

  $s_{127} = \{\ r_{1271}(C_{90}, C_{70})\ \}$

  $s_{128} = \{\ r_{1281}(C_{91}, C_{71})\ \}$

  $s_{129} = \{\ r_{1291}(C_{72}, C_{91})\ \}$

  $s_{130} = \{\ r_{1301}(C_{90}, C_{91})\ \}$

  $s_{131} = \{\ r_{1311}(C_{91}, C_{92}), r_{1312}(C_{91}, C_{90})\ \}$

  $s_{132} = \{\ r_{1321}(C_{70}, C_{71}), r_{1322}(C_{70}, C_{91}), r_{1323}(C_{92}, C_{91}), r_{1324}(C_{92}, C_{71})\ \}$

  $s_{133} = \{\ r_{1331}(C_{71}, C_{70})\ \}$

  $s_{134} = \{\ r_{1341}(C_{90}, C_{91})\ \}$

  $s_{135} = \{\ r_{1351}(C_{91}, C_{90}), r_{1352}(C_{91}, C_{92})\ \}$

  $s_{136} = \{\ r_{1361}(C_{92}, C_{91})\ \}$

  $\}$
  of 29 sequential binary relations.

- *The set $S_{12} = \{s_{123}, s_{124}, s_{125}, s_{126}, s_{127}, s_{128}, s_{129}, s_{130}, s_{131}, s_{132}, s_{133}, s_{134}, s_{135}, s_{136}\}$ of 14 discernible states.*

- *The set* $C_{12} = \{C_{70} = \{(x_7, \delta_{70})\}, C_{71} = \{(x_7, \delta_{71})\}, C_{72} = \{(x_7, \delta_{72})\}, C_{90} = \{(x_9, \delta_{90})\}, C_{91} = \{(x_9, \delta_{91})\}, C_{92} = \{(x_9, \delta_{92})\}\}$ *of 6 observations classes.*

The figure 6.18 provides a graphical representation of $BM(X_{12}(t))$. This behavioral model is composed of three behavioral models: one that characterizes the drain behaving as a pipe (ϕ_{210}) which is the normal behavior, another characterizes the drain behaving as a low pass filter (ϕ_{211}) and a last one characterizing the drain behaving as a tank (ϕ_{212}).

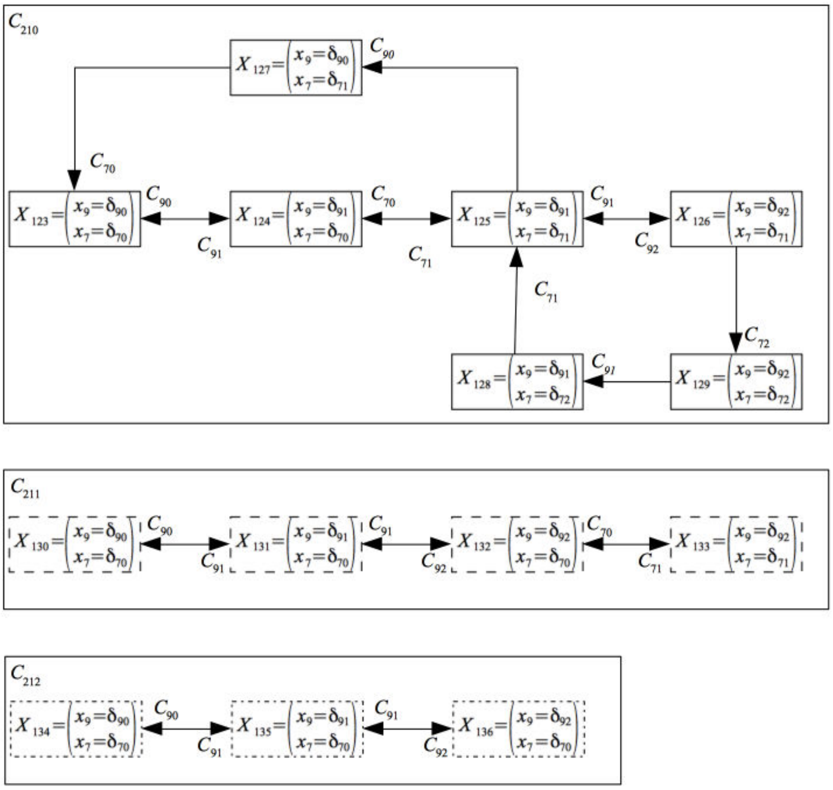


Figure 6.18: Generic Behavioral Model $BM(X_{12}(t))$

The model $M(X_{12}(t))$ of the dynamic process $X_{12} = \{x_8, x_9, \phi_{21}\}$ is the tuple $M(X_{12}(t)) =< PM(X_{12}(t)), SM(X_{12}(t)), FM(X_{12}(t)), BM(X_{12}(t)) >$.

6.5.6 Step 3.3: Generic Modeling of $X_{13}(t)$

The Perception Model of the process $X_{13}(t)$ is given in the definition 6.12:

Definition 6.12 *The Perception Model* $PM(X_{13}(t))$ *of the process* $X_{13}(t) = \{x_7(t), x_9(t), x_{10}(t), x_{11}(t)\}$ *is the structure* $< X_{13}, \Psi_{13}, R^{q_{13}} >$ *where:*

- $X_{13} = \{x_7(t), x_9(t), x_{10}(t), x_{11}(t)\}$,
- $\Psi = \{\Psi_{91}, \Psi_{92}\, \Psi_{71}, \Psi_{72}\, \Psi_{100}, \Psi_{110}\}$
- $R^{q_{13}} = R^{goal_{13}} \cup R^{n_{13}} \cup R^{ab_{13}}$.
  - $R^{goal_{13}} = \{\Phi\}$ *(no goal can be defined:* $X_{13}(t)$ *must work normally)*
  - $R^{n_{13}} = \{\forall t, x_{10}(t) < \Psi_{101} \wedge x_{11}(t) < \Psi_{111}\}$

- $R^{ab_{13}} = \{\forall t, x_{10}(t) \geq \Psi_{101} \vee x_{11}(t) \geq \Psi_{111}\}$

The Structural Model of $X_{13}(t)$ is given in the definition 6.13 and is represented in the figure 6.19.

Definition 6.13 *The structural model $SM(X_{13}(t))$ is the structure $< COMPS_{13}, R^{p_{13}}, R^{x_{13}} >$ where:*

- $COMPS_{13} = \{c_3, c_1, c_4, c_9, c_7, c_{10}\}$
- $R^{p_{13}} = \{out(c_3) = in_1(c_4), out(c_1) = in_2(c_4)\}$
- $R^{x_{13}} = \{x_9 = out(c_9), x_7 = out(c_7), x_{10} = out_1(c_{10}), x_{11} = out_2(c_{10})\}$

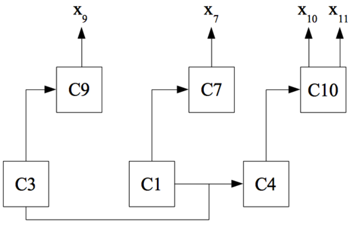


Figure 6.19: Generic Structural Model $SM(X_{13(t)})$

The Generic Functional Model $FM(X_{13}(t))$ of $X_{13}(t)$ is given in the definition 6.14 (cf. figure 6.20.

Definition 6.14 *The Functional Model $FM(X_{13}(t))$ of the hydraulic process $X_{13}(t)$ is the structure $< \Delta_{13}, F_{13}, R^{f_{13}} >$ where,*

- $\Delta_{13} = \Delta_{x_9} \cup \Delta_{x_7} \cup \Delta_{x_{10}} \cup \Delta_{x_{11}}$.
- $F_{13} = \{f_3 : \Delta^{x_9} \times \Delta^{x_7}; f_4 : \Delta^{x_9} \times \Delta^{x_7}\}$.
- $R^f = \{x_{10} = f_3(x_9, x_7); x_{11} = f_3(x_9, x_7)\}$.

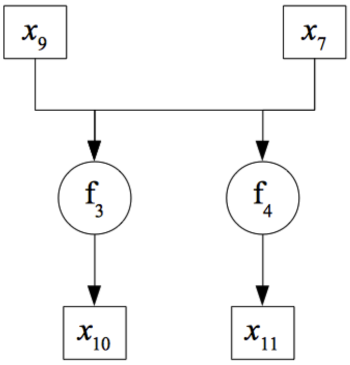


Figure 6.20: Generic Functional Model $FM(X_{13}(t))$

The process $X_{13}(t)$ being a purely combinatorial process, it is completely described with the table 6.9. As a consequence, $X_{13}(t)$ does not need a behavioral model: the relations between the values of x_7 and x_9 and those of x_{10} and x_{11} will be directly computed with the table of values of f_3 and $f4$.

6.6 Diagnosis

6.6.1 Main elements

The behavioral model $BM(X(t))$ of $X(t) = X_{11}(t) \cup X_{12}(t) \cup X_{13}(t)$ is the union of the behavioral models of its sub-processes $BM(X(t)) = BM(X_{11}(t)) \cup BM(X_{12}(t))$.

The perception model of $X(t) = X_{11}(t) \cup X_{12}(t) \cup X_{13}(t)$ is the union $PM(X_{11}(t) \cup PM(X_{12}(t)) \cup PM(X_{13}(t))$:

- $R^{goal_{11}} = \{\exists t_0, \exists t_i, t_i \geq t_0,$
  $\forall t \geq t_i, x_8(t) = \delta_{81}$
  $\}$

- $R^{n_{11}} = \{\forall t \geq t_i,$
  $x_8(t) = \delta_{81} \wedge$
  $x_9(t) = \delta_{91} \wedge$
  $\}$

- $R^{ab_{11}} = \{\forall t \geq t_i,$
  $x_8(t) = \delta_{80} \vee x_8(t) = \delta_{82} \vee$
  $x_9(t) = \delta_{90} \vee x_9(t) = \delta_{92} \vee$
  $\}$

- $R^{n_{12}} = \{\forall t \geq t_i,$
  $x_9(t) = \delta_{91} \wedge$
  $x_7(t) = \delta_{71}$
  $\}$

- $R^{ab_{12}} = \{\forall t \geq t_i,$
  $x_9(t) = \delta_{90} \vee x_9(t) = \delta_{92} \vee$
  $x_7(t) = \delta_{70} \vee x_7(t) = \delta_{72} \vee$
  $\}$

- $R^{n_{13}} = \{\forall t,$
  $x_{10}(t) = \delta_{102} \wedge x_{11}(t) = \delta_{112}$
  $\}$

- $R^{n_{13}} = \{\forall t,$
  $(x_9(t) = \delta_{90} \wedge x_7(t) = \delta_{70}) \vee$
  $(x_9(t) = \delta_{91} \wedge x_7(t) = \delta_{71}) \vee$
  $(x_9(t) = \delta_{92} \wedge x_7(t) = \delta_{72})$
  $\}$

- $R^{ab_{13}} = \{\forall t,$
  $x_{10}(t) = \delta_{100} \vee x_{11}(t) = \delta_{110} \vee$
  $x_{10}(t) = \delta_{101} \vee x_{11}(t) = \delta_{111} \vee$
  $\}$

- $R^{ab_{13}} = \{\forall t,$
  $(x_9(t) = \delta_{91} \wedge x_7(t) = \delta_{70})\vee$
  $(x_9(t) = \delta_{92} \wedge x_7(t) = \delta_{70})\vee$
  $(x_9(t) = \delta_{92} \wedge x_7(t) = \delta_{71})$
  $\}$

6.6.2 Operating goals and Unsatisfactory States

Figures 6.21 and 6.22 exhibit the graphical representation of the observation models for $X_{11}(t)$ and $X_{12}(t)$. Let us recall that $OM(X_{11}(t))$ is composed by two sub-models, one corresponding to the case the Upstream Shoulder acts as a Low Pass Filter (normal functioning) and one corresponding to the case it acts as a Pipe (faulty functioning). In the same way, $OM(X_{12}(t))$ is composed by three sub-models corresponding respectively to the case the Vertical Drain acts as a Pipe, resp. a Low Pass Filter, resp. a Tank.

To define if the current state $s(t_k)$ of the Sapins' Dam process is unsatisfactory, it is necessary to compare $s(t_k)$ with the propositions of R^{ab}. To this aim, we build a set S^{ab} containing the undesirable states of $X(t)$ according to the following rules:

- **Rule 1**: $\forall t_k, \forall s \in S, \exists t_i, t_i \geq t_0 \wedge t_k > t_i \wedge \exists r(C_\phi, C_{80}), [\phi, \phi]) \in s \rightarrow s \in S^{ab}(t_k)(X_{11}(t))$
- **Rule 2**: $\forall t_k, \forall s \in S, \exists t_i, t_i \geq t_0 \wedge t_k > t_i \wedge \exists r(C_\phi, C_{82}), [\phi, \phi]) \in s \rightarrow s \in S^{ab}(t_k)(X_{11}(t))$
- **Rule 3**: $\forall t_k, \forall s \in S, \exists t_i, t_i \geq t_0 \wedge t_k > t_i \wedge \exists r(C_\phi, C_{90}), [\phi, \phi]) \in s \rightarrow s \in S^{ab}(t_k)(X_{11}(t))$
- **Rule 4**: $\forall t_k, \forall s \in S, \exists t_i, t_i \geq t_0 \wedge t_k > t_i \wedge \exists r(C_\phi, C_{92}), [\phi, \phi]) \in s \rightarrow s \in S^{ab}(t_k)(X_{11}(t))$
- **Rule 5**: $\forall t_k, \forall s \in S, \exists t_i, t_i \geq t_0 \wedge t_k > t_i \wedge \exists r(C_\phi, C_{90}), [\phi, \phi]) \in s \rightarrow s \in S^{ab}(t_k)(X_{12}(t))$
- **Rule 6**: $\forall t_k, \forall s \in S, \exists t_i, t_i \geq t_0 \wedge t_k > t_i \wedge \exists r(C_\phi, C_{92}), [\phi, \phi]) \in s \rightarrow s \in S^{ab}(t_k)(X_{12}(t))$
- **Rule 7**: $\forall t_k, \forall s \in S, \exists t_i, t_i \geq t_0 \wedge t_k > t_i \wedge \exists r(C_\phi, C_{70}), [\phi, \phi]) \in s \rightarrow s \in S^{ab}(t_k)(X_{12}(t))$
- **Rule 8**:$\forall t_k, \forall s \in S, \exists t_i, t_i \geq t_0 \wedge t_k > t_i \wedge \exists r(C_\phi, C_{72}), [\phi, \phi]) \in s \rightarrow s \in S^{ab}(t_k)(X_{12}(t))$

The use of the $OM(X(t))$ models make easier the detection of abnormal states with an *a priori* identification of the corresponding relations:

- **Rule 1** $(OM(X_{11}(t)))$: r_{1151}, r_{1153}, r_{1161}, r_{1165}, r_{1211}.
- **Rule 2** $(OM(X_{11}(t)))$: r_{1141}, r_{1144}, r_{1152}, r_{1154}, r_{1162}, r_{1166}, r_{1213}.
- **Rule 3** $(OM(X_{11}(t)))$: r_{1121}, r_{1124}, r_{1163}, r_{1167}.
- **Rule 4** $(OM(X_{11}(t)))$: r_{1112}, r_{1164}, r_{1168}, r_{1202}, r_{1212}, r_{1222}.
- **Rule 5** $(OM(X_{12}(t)))$: r_{1242}, r_{1244}, r_{1253}, r_{1255}, r_{1312}, r_{1351}.
- **Rule 6** $(OM(X_{12}(t)))$: r_{1252}, r_{1254}, r_{1311}, r_{1352}.
- **Rule 7** $(OM(X_{12}(t)))$: r_{1251}, r_{1256}, r_{1271}, r_{1331}.
- **Rule 8** $(OM(X_{12}(t)))$: r_{1262}.

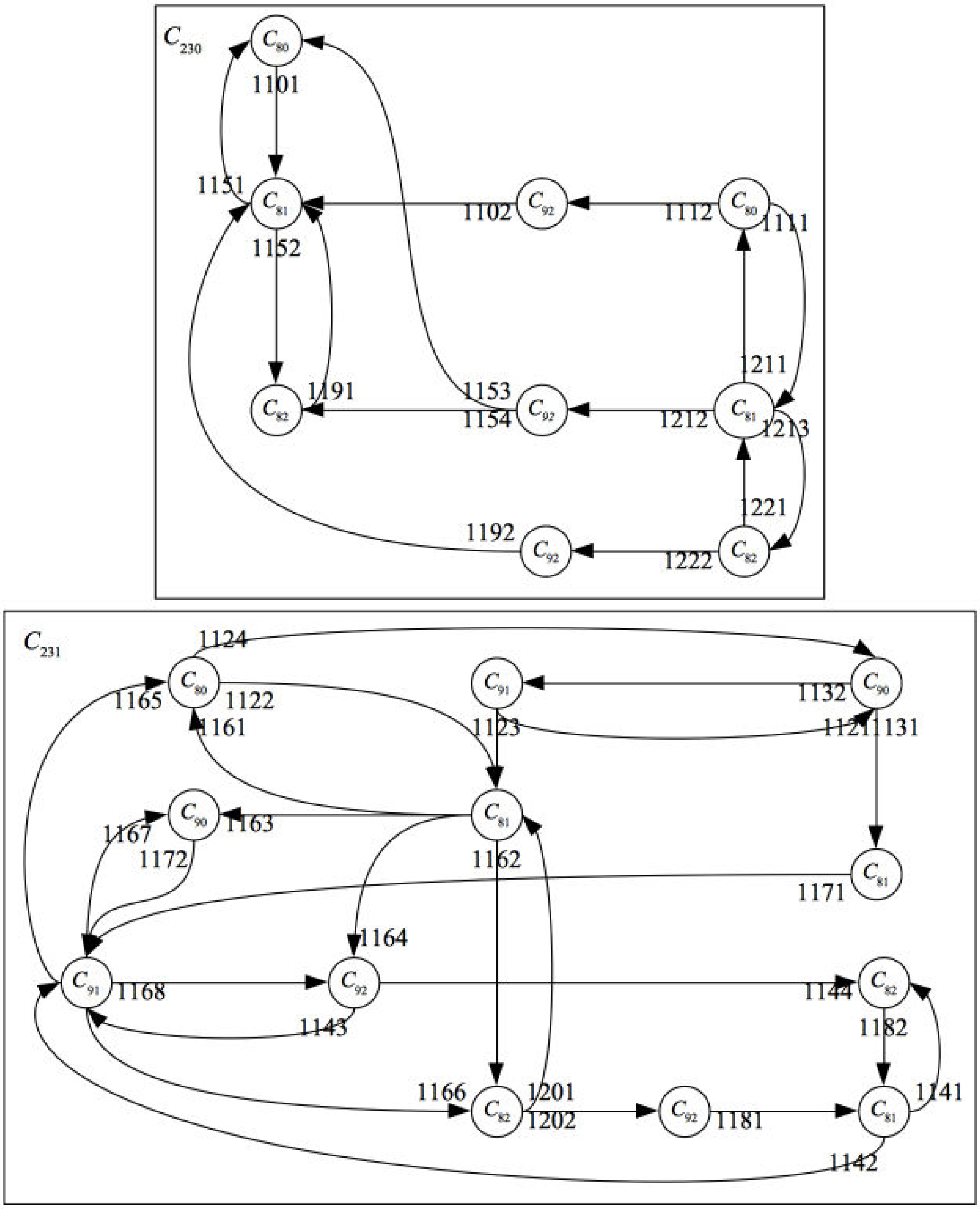


Figure 6.21: Observation Model $OM(X_{11}(t))$

This leads to define the following set of relations corresponding to abnormal states:

- $R^{ab}_{dynamic} = \{$ $r_{1112}, r_{1121}, r_{1124}, r_{1141}, r_{1144}, r_{1151}, r_{1152}, r_{1153}, r_{1154}, r_{1161}, r_{1162}, r_{1163}, r_{1164}, r_{1165}, r_{1166}, r_{1167}, r_{1168}, r_{1202}, r_{1211}, r_{1212}, r_{1213}, r_{1222}, r_{1242}, r_{1244}, r_{1251}, r_{1252}, r_{1253}, r_{1254}, r_{1255}, r_{1256}, r_{1262}, r_{1271}, r_{1311}, r_{1312}, r_{1331}, r_{1351}, r_{1352}.$ $\}$

6.6.3 Diagnosis of the Sapins Dam phenomena

We assume the following hypothesis:

1. The initial state is unknown.

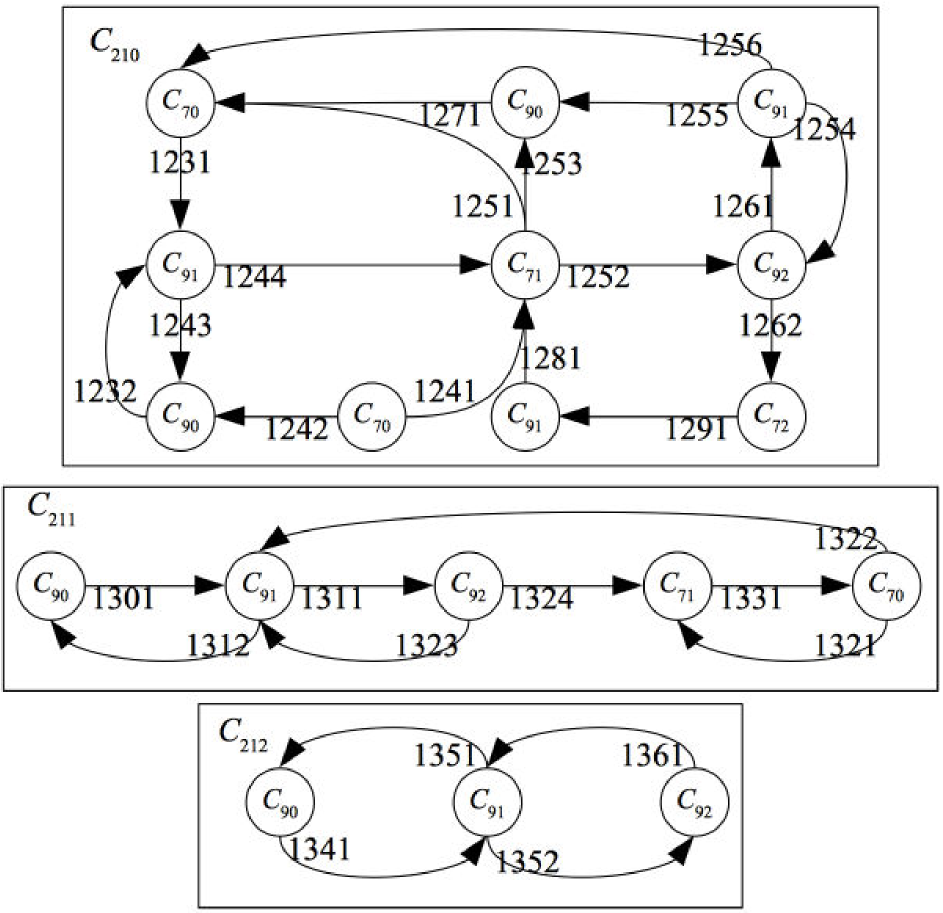


Figure 6.22: Observation Model $OM(X_{12}(t))$

2. At the end of each sequence, the process is supposed to be in an unsatisfactory state that justifies the request to a diagnosis.

The scenario that occurs for the Sapins' Dam is the sequence ω containing the following observation occurrences:

1. $C_{91}(11/1978)$: $x_9(11/1978) = \delta_{91}$ (the pressure cell is Medium).
2. $C_{81}(12/1978)$: $x_8(12/1978) = \delta_{81}$ (the level of the water in the reservoir is Medium).
3. $C_{92}(04/1979)$: $x_9(04/1979) = \delta_{92}$ (the pressure cell is High).
4. $C_{71}(11/1980)$: $x_7(11/1980) = \delta_{71}$ (the drain flow is Medium).
5. $C_{70}(11/1981)$: $x_7(11/1981) = \delta_{70}$ (the drain flow is Low).
6. $C_{80}(11/1981)$: $x_8(11/1981) = \delta_{80}$ (the level of the water in the reservoir is Low).
7. $C_{71}(12/1981)$: $x_7(12/1981) = \delta_{71}$ (the drain flow is Medium).
8. $C_{81}(12/1981)$: $x_8(12/1981) = \delta_{81}$ (the level of the water in the reservoir is Medium).
9. $C_{70}(02/1983)$: $x_7(02/1983) = \delta_{70}$ (the drain flow is Low).
10. $C_{80}(12/1985)$: $x_8(12/1985) = \delta_{80}$ (the level of the water in the reservoir is Low).

11. $C_{81}(03/1986)$: $x_8(03/1986) = \delta_{81}$ (the level of the water in the reservoir is Medium).

12. $C_{110}(09/1988)$: $x_{11}(09/1988) = \delta_{112}$ (a wet area is observable on the downstream shoulder).

13. $C_{100}(10/1988)$: $x_{10}(10/1988) = \delta_{102}$ (a leakage is observable on the downstream shoulder).

The first important idea of the diagnosis algorithm is to *explain* the faults with occurrences $C_i(t_k)$ of observation classes that are not linked with sensors. The challenge is to find the *unobserved* classes that could explain the current state of the Sapins Dam process and to place them in a set of paths describing the way the process reached an unsatisfactory state. To this aim we use the observation model $OM(X(t))$ in place of the behavioral model $BM(x(t))$ to build the paths.

These paths are used to build a sequence of state vector values ..., $X_i(t_k))$, ... that can be used to identify, if required, the corresponding states. We distinguish five possible paths extracted from the five observation sub-models: P_{11-a}, P_{11-b}, P_{12-a}, P_{12-b} and P_{12-c}.

The execution of the $TOM4E$ algorithm leads to the following traces:

1. $C_{91}(11/1978)$: $x_9(11/1978) = \delta_{91}$
   The initial time is unknown but lesser or equal to 11/1978, no relation can be identified.
   $X(11/1978) = \{\ x_7 = \emptyset,\ x_8 = \emptyset,\ x_9 = \delta_{91},\ x_{10} = \emptyset,\ x_{11} = \emptyset,\ \phi_{21} = \emptyset,\ \phi_{23} = \emptyset\ \}$

2. $C_{81}(12/1978)$: $x_8(12/1978) = \delta_{81}$
   At $t = 12/1978$, *getRelations*(C_{91}, C_{81}) applied on $OM_{11}(X(t))$ returns r_{1123} which is satisfactory: the goal is reached.
   $P_{11-a} = \{\ p_{111} = \{r_{1123}\}\ \}$
   P_{12} is not concerned by $C_{81}(12/1978)$
   $X(12/1978) = \{\ x_7 = \emptyset,\ x_8 = \delta_{81},\ x_9 = \delta_{91},\ x_{10} = \emptyset,\ x_{11} = \emptyset,\ \phi_{21} = \emptyset,\ \phi_{23} = \emptyset\ \}$

3. $C_{92}(04/1979)$: $x_9(04/1979) = \delta_{92}$
   At $t = 04/1979$, *getRelations*(C_{81}, C_{92}) applied on $OM_{11}(X(t))$ returns two relations : r_{1164} and r_{1212} As the goal had been reached since $t = 12/1978$, these relations are unsatisfactory. The paths in P_{11-a} can be updated according to the function $isAPath(p_i, r_{1164}) = true$.
   $P_{11-a} = \{\ p_{111} = \{r_{1123}, r_{1164}\}\ \}$
   $P_{11-b} = \{\ p_{112} = \{r_{1212}\}\ \}$
   A path p_{112} is initiated for $OM(X_{11-b}))$ which corresponds to an abnormal functioning of the Upstream Shoulder. The other path p_{111} corresponds to a normal functioning. Indeed, at this stage, we can not infer a faulty behaviour.
   Applied on $OM_{12}(X(t))$, *getRelations*(C_{91}, C_{92}) returns two relations: r_{1254}, r_{1311} and r_{1352}. These relations are unsatisfactory.
   $P_{12} = \{\ p_{121} = \{r_{1254}\},\ p_{122} = \{r_{1311}\},\ p_{123} = \{r_{1352}\}\ \}$
   It is to note that a path p_{122} is initiated for $OM(X_{12-b})$ and one p_{123} for $OM(X_{12-c})$which correspond to abnormal functioning of the Vertical Drain while the other path p_{121} corresponds to a normal functioning. Indeed, at this stage, we can not infer a faulty behaviour.
   $X(04/1979) = \{\ x_7 = \emptyset,\ x_8 = \delta_{81},\ x_9 = \delta_{92},\ x_{10} = \emptyset,\ x_{11} = \emptyset,\ \phi_{21} = \emptyset,\ \phi_{23} = \emptyset\}$

4. $C_{71}(11/1980)$: $x_7(11/1980) = \delta_{71}$
   P_{11} is not concerned by $C_{71}(11/1980)$
   At $t = 11/1980$, $getRelations(C_{92}, C_{71})$ applied on $OM_{12}(X(t))$ returns one relation r_{1324}:
   The path in P_{12} can be updated according to the function $isAPath(p_i, r_{1234}) = true$. Only p_{122} is kept, p_{121} and p_{123} are removed
   $P_{12} = \{\ p_{122} = \{r_{1311}, r_{1324}\}\ \}$
   We can now infer that the Vertical Drain has a faulty behaviour: it acts as a Low Pass Filter.
   Using Table 6.9, it can be deduced that $x_{10} = \delta_{101}$ and $x_{11} = \delta_{111}$: a leakage is suspected in the Downstream Shoulder.
   $X(11/1980) = \{\ x_7 = \delta_{71},\ x_8 = \delta_{81},\ x_9 = \delta_{92},\ x_{10} = \delta_{101},\ x_{11} = \delta_{111},\ \phi_{21} = \delta_{211},\ \phi_{23} = \emptyset\}$

5. $C_{70}(11/1981)$: $x_7(11/1981) = \delta_{70}$
   P_{11} is not concerned by $C_{70}(11/1981)$
   At $t = 11/1981$, $getRelations(C_{71}, C_{70})$ applied on $OM_{12}(X(t))$ returns one relations r_{1331}
   The path in P_{12} can be updated according to the function $isAPath(p_i, r_{1331}) = true$. r_{1331} is an unsatisfactory relation.
   $P_{12} = \{\ p_{122} = \{r_{1311}, r_{1324}, r_{1331}\}\ \}$
   Using Table 6.9, it can be deduced that $x_{10} = \delta_{102}$ and $x_{11} = \delta_{112}$: a leakage is confirmed in the Downstream Shoulder.
   $X(11/1981) = \{\ x_7 = \delta_{70},\ x_8 = \delta_{81},\ x_9 = \delta_{92},\ x_{10} = \delta_{102},\ x_{11} = \delta_{112},\ \phi_{21} = \delta_{211},\ \phi_{23} = \emptyset\ \}$.

6. $C_{80}(11/1981)$: $x_8(11/1981) = \delta_{80}$
   At $t = 11/1981$, $getRelations(C_{92}, C_{80})$ applied on $OM_{11}(X(t))$ returns one relation: r_{1153}.
   The path in P_{11} can be updated according to the function $isAPath(p_i, r_{1153}) = true$. The path p_{111} corresponding to the situation the Upstream Shoulder acts as a Low Pass Filter is no more kept. This leads us to conclude that the Upstream Shoulder gets from this time a behaviour of pipe (δ_{230})
   $P_{11-b} = \{\ p_{112} = \{r_{1212}, r_{1153})\}\ \}$
   P_{12} is not concerned by $C_{80}(11/1981)$
   P_{13} is not concerned by $C_{80}(11/1981)$
   $X(11/1981) = \{\ x_7 = \delta_{70},\ x_8 = \delta_{80},\ x_9 = \delta_{92},\ x_{10} = \delta_{102},\ x_{11} = \delta_{112},\ \phi_{21} = \delta_{211},\ \phi_{23} = \delta_{230}\ \}$

7. $C_{71}(12/1981)$: $x_7(12/1981) = \delta_{71}$
   P_{11} is not concerned by $C_{71}(12/1981)$
   At $t = 12/1981$, $getRelations(C_{70}, C_{71})$ applied on $OM_{12}(X(t))$ returns one relation r_{1321}
   The path in P_{12} can be updated according to the function $isAPath(p_i, r_{1321}) = true$. $P_{12} = \{\ p_{122} = \{r_{1311}, r_{1324}, r_{1331}, r_{1321}\}\ \}$
   Using Table 6.9, it can be deduced that $x_{10} = \delta_{101}$ and $x_{11} = \delta_{111}$: a leakage is suspected in the Downstream Shoulder. $X(12/1981) = \{\ x_7 = \delta_{71},\ x_8 = \delta_{80},\ x_9 = \delta_{92},\ x_{10} = \delta_{101},\ x_{11} = \delta_{111},\ \phi_{21} = \delta_{211},\ \phi_{23} = \delta_{230}\}$

8. $C_{81}(12/1981)$: $x_8(12/1981) = \delta_{81}$
   At $t = 12/1981$, $getRelations(C_{80}, C_{81})$ applied on $OM_{11}(X(t))$ returns one relations r_{1101}.

The path in P_{11} can be updated according to the function $isAPath(p_i, r_{1101}) = true$. The path is updated: $P_{11-b} = \{\ p_{112} = \{r_{1212}, r_{1153}, r_{1101}\}\ \}$
P_{12} is not concerned by $C_{81}(12/1981)$
P_{13} is not concerned by $C_{81}(12/1981)$
$X(12/1981) = \{\ x_7 = \delta_{71},\ x_8 = \delta_{81},\ x_9 = \delta_{92},\ x_{10} = \delta_{101},\ x_{11} = \delta_{111},\ \phi_{21} = \delta_{211},\ \phi_{23} = \delta_{230}\}$

9. $C_{70}(02/1983)$: $x_7(02/1983) = \delta_{70}$
P_{11} is not concerned by $C_{71}(02/1983)$
At $t = 02/1983$, $getRelations(C_{71}, C_{70})$ applied on $OM_{12}(X(t))$ returns one relation r_{1331}
The path in P_{12} can be updated according to the function $isAPath(p_i, r_{1331}) = true$. $P_{12} = \{\ p_{122} = \{r_{1311}, r_{1324}, r_{1331}, r_{1321}, r_{1331}\}\ \}$
Using Table 6.9, it can be deduced that $x_{10} = \delta_{101}$ and $x_{11} = \delta_{111}$: a leakage is confirmed in the Downstream Shoulder. $X(02/1983) = \{\ x_7 = \delta_{70},\ x_8 = \delta_{81},\ x_9 = \delta_{92},\ x_{10} = \delta_{102},\ x_{11} = \delta_{112},\ \phi_{21} = \delta_{211},\ \phi_{23} = \delta_{230}\}$

10. $C_{80}(12/1985)$: $x_8(12/1985) = \delta_{80}$
At $t = 12/1985$, $getRelations(C_{81}, C_{80})$ applied on $OM_{11}(X(t))$ returns one relation r_{1151}.
The path in P_{11} can be updated according to the function $isAPath(p_i, r_{1151}) = true$. The path is updated: $P_{11-b} = \{\ p_{112} = \{r_{1212}, r_{1153}, r_{1101}, r_{1151}\}\ \}$
P_{12} is not concerned by $C_{81}(12/1985)$
P_{13} is not concerned by $C_{81}(12/1985)$
$X(t_{12/1985}) = \{\ x_7 = \delta_{70},\ x_8 = \delta_{80},\ x_9 = \delta_{92},\ x_{10} = \delta_{102},\ x_{11} = \delta_{112},\ \phi_{21} = \delta_{211},\ \phi_{23} = \delta_{230}\}$

11. $C_{81}(03/1986)$: $x_8(03/1986) = \delta_{81}$
At $t = 03/1986$, $getRelations(C_{80}, C_{81})$ applied on $OM_{11}(X(t))$ returns one relations r_{1101}.
The path in P_{11} can be updated according to the function $isAPath(p_i, r_{1101}) = true$. The path is updated: $P_{11-b} = \{\ p_{112} = \{r_{1212}, r_{1153}, r_{1101}, r_{1151}, r_{1101}\}\ \}$
P_{12} is not concerned by $C_{81}(03/1986)$
P_{13} is not concerned by $C_{81}(03/1986)$
$X(03/1986) = \{\ x_7 = \delta_{70},\ x_8 = \delta_{81},\ x_9 = \delta_{92},\ x_{10} = \delta_{102},\ x_{11} = \delta_{112},\ \phi_{21} = \delta_{211},\ \phi_{23} = \delta_{230}\ \}$

12. $C_{112}(09/1988)$: $x_{11}(09/1988) = \delta_{112}$
P_{11} is not concerned by $C_{112}(09/1988)$
P_{12} is not concerned by $C_{112}(09/1988)$
$x_{11} = \delta_{112}$ is here confirmed by a real measurement performed thanks to a visual observation. The algorithm has detected this situation since 11/1981.
$X(09/1988) = \{\ x_7 = \delta_{70},\ x_8 = \delta_{81},\ x_9 = \delta_{92},\ x_{10} = \emptyset,\ x_{11} = \delta_{112},\ \phi_{21} = \delta_{211},\ \phi_{23} = \delta_{230}\ \}$

13. $C_{102}(10/1988)$: $x_{10}(10/1988) = \delta_{102}$
P_{11} is not concerned by $C_{102}(10/1988)$
P_{12} is not concerned by $C_{102}(10/1988)$
As for x_{11}, $x_{10} = \delta_{102}$ is here confirmed by a real measurement performed thanks to a visual observation. The algorithm has detected this situation since 11/1981.

$X(10/1988) = \{\ x_7 = \delta_{70},\ x_8 = \delta_{81},\ x_9 = \delta_{92},\ x_{10} = \delta_{102},\ x_{11} = \delta_{112},\ \phi_{21} = \delta_{211},$ $\phi_{23} = \delta_{230}\}$

6.7 Interpretation of the results

The first undesirable state is encountered when $x_9(04/1979) = \delta_{92}$ (*PressureCell* is *High*).

In 11/1980, the algorithm detected that the drain is in an unsatisfactory state: the behavior of the Vertical Drain has changed from a Pipe to a Low Filter Pass since this moment. Moreover, later in 11/1981), the diagnosis let us know, without any doubt, that the Upstream Shoulder changed of component type: since this date, it has behaved as a Pipe and no more as a Low Pass Filter that is its normal functioning behavior. The detection of leakage in the Downstream Shoulder is very early, as in 11/1980, the algorithm concludes that there is water present in that component while this will be confirmed by a visual observation in 09/1988, indeed 8 years later...

These findings are totally consistent with the explanations given by the experts after the failure of the dam [PRB06]. The moments they are detected are very early in the scenario (see for instance later the date at which the algorithm detects the leakage and the timed explanations given *a posteriori* by the experts). Before mapping the results given by $TOM4E$ and the expert explanation, let give an extract of the expert analysis: *Investigations at the Sapins dam concluded that a mechanism of internal erosion was operative. The particle size grading of the fill material made it particularly sensitive to internal erosion, and this led to the gradual clogging of the vertical drain aided by the fact that the drain did not meet standard filter rules. Clogging caused first the upstream shoulder to become gradually saturated, then the downstream shoulder as the seepage over-topped the drain (the top was lower than the normal reservoir level).*

Indeed, the clogging of the vertical drain is perfectly consistent with the increase of the drain resistance and conversely, the internal erosion of the upstream shoulder that leads to a loss of the shoulder material is equally consistent with the decrease of the shoulder resistance. Concerning now the presence of water in the downstream shoulder, the experts considered that it began to occurred in 11/1980 when the pressure cell assessed by c_3 got a value over than the height of the drain and the water present in the downstream shoulder passed over the drain. Those results are perfectly consistent with our findings.

This makes the demonstration of the interest of the algorithm coupled with TOM4D modeling: in November 1980, the first problem would have been identified using TOM4D models and TOM4E algorithm, eight years before the quasi-failure of the dam. The presence of water that is a very dangerous situation for an embankment dam is highlighted, for itself, in November 1981, seven years before the quasi-failure.

TOM4D models and the TOM4E algorithm are then consequently powerful tools to detect and explain the reasons that led to the Sapins Dam behavior deterioration.

CHAPTER 7

Conclusion

Modeling process is crucial in system diagnosis stage, which represents the source knowledge for diagnosis algorithm. However some systems, like Dam, are characterized by the multiplicity of the behaviors and the heterogeneity of the system components, the temporal feature and knowledge of different natures and sources: visual observation (Experts) and program observation (Sensors).

7.1 Thesis Overview

The main challenge is to define a diagnosis approach that takes into account all these features. This task is developed, in this thesis, by the means of a number of intermediate stages.

First, we have defined a modeling approach for diagnosis of systems that fall in the area of knowledge Expert : TOM4D (Timed Observation Modeling for Diagnosis). TOM4D is Multi-modeling approach based on the Theory of Timed Observations, a mathematical framework for modeling and reasoning about dynamic process from timed data. The proposed TOM4D methodology is a primarily syntax-driven modeling approach where semantic content is introduced in a gradual and controlled way through the CommonKADS conceptual approach and the Tetrahedron of States. TOM4D acts as a bridge between KE and KDD allowing to build a process model which, by construction, can be directly related on the knowledge model provided by the expert, and besides, it can be collated with models obtained from data.

Our second contribution concerns the use of the models obtained to carry out a diagnosis by designing an algorithm dedicated to the automatic identification of: (i) the phenomena involved in the deterioration of dam reliability and safety; and (ii) the causes of these deterioration. Our proposed approach makes several contributions to the diagnosis problem. First, the TOM4D approach is based on modeling with the same level that the expert uses and the system observation is based on the Theory of timed observation using the observation class (which gives a meaning to the event). Adding a semantic meaning to the observations and the models avoid us to check the diagnosability of the system, since we treat only the observed and real (physical) observations. Second, we provide a new framework with a right level of abstraction in which the models have to be constructed to obtain an efficient diagnosis by considering only the components that are concerned with diagnosing and thus, the number of components are minimal allowing a more efficient diagnosis. Our algorithm is optimized in the sense that with the abstracted information, the search space is reduced to be as small as possible. Finally, the interpretation of the results is based on the perception model in order to identify the diagnosis objectives.

Then, we propose a new framework for diagnosis where each component has only the access to its own observable observations according to the Decomposition property of the Timed Observation Theory. Therefore, we extended the TOM4D methodology to cope with this decomposition property. This leads to consider each component as a sub-system apart with a behavior independent from each other. the communication between component behavior model is provided by the relations between variables defined in the global functional and structural models. The diagnosis algorithm enable to compute the global diagnosis without necessarily building global objects, whose construction can be very computationally costly for large and complex systems.

Finally, the proposed approach of diagnosis was applied to the dam diagnosis issue. We have shown, on one hand, the advantage of the TOM4D approach to represent a complex and dynamic system like the dam. The models obtained were used to carry out a diagnosis by applying the proposed algorithm. This step is dedicated to the automatic identification of the phenomena involved in the deterioration of dam reliability and safety. The obtained results has been validated by the french Sapins dam experts.

The results presented in the previous chapters indicate that these goals have been met and that the overall objective of the thesis has been achieved.

7.2 Synthesis

Generally speaking, fault assessment is a knowledge intensive task that requires a model of the process under consideration: to diagnose a process consists in a reasoning aiming at explaining the available *observations* with a *model* of the system according to an adequate *problem solving method*. The possibility for a computer to achieve such a diagnosis reasoning depends on the manner*observations*, *models* and *problem solving methods* are formalized.

The State of the Art chapter of this document presents the basis of the three main categories of diagnosis approaches (DX, FDI and DES for short), with the didactic example of an hydraulic system described in [CPR00]. The aim of this chapter is to elicit the common concepts and the specificities of each of these main approaches, and to emphasis on the particular difficulties when diagnosing complex dynamic system where the temporal aspects play a crucial role. It introduces also the basis of the Multi-Modeling Based Diagnosis (MMBD) as an attempt to deal with complex dynamic processes. The TOM4D modeling methodology (Timed Observation Modeling for Diagnosis) is a new MMBD approach, based on the Timed Observation Theory (TOT) proposed by Le Goc in 2006 [LG06], where the timestamps of the data play a central role.

This mathematical theory combines and extends the Markov Chain and the Poisson Process Theories [CL99], the Theory of Communication of Shannon [Sha84], the Logical Theory of Diagnosis [Rei87] and the Method of Abstraction of Floridi [Flo08]. Up to our knowledge, the TOT is the only mathematical theory that has been designed to propose a unique framework to model networks of continuous time dynamic processes at different levels of abstraction and granularity.

According to the Timed Observations Theory, the notion of fault is concerned with particular behaviors of a dynamic process: a fault is linked with the occurrence of an undesired behavior of an observed process. This means that the required knowledge to assess a fault about a dynamic process is the one required by the tasks of monitoring, diagnosis and prognosis. The quality of the knowledge corpus required by these tasks is directly linked with the pertinence of the fault assessment task. Five modeling principles have been derived from the mathematical tools of the TOT in order to constitutes a strong logical basis for the TOM4D methodology: from the identification of the variables, the knowledge engineer will identify the possible values a variable can take over time, its corresponding function and observation classes and so, it defines the discernible state space of an observed dynamic process. Next, the knowledge analysis examines all the possible and the impossible relations between two elements (variable, constant, component and observation classes), conducted through their semantic properties.

The organization of the resulting knowledge representations in the four models leads to an operational model of the dynamic system.

The main advantages of the TOM4D modeling methodology are the followings:

- Abstraction level.
  The TOM4D model resides at the same abstraction level as the expert's reasoning. The TOM4D models are a representation of the way an expert *perceive* a dynamic process.

- Compatibility with the available data.
  TOM4D being based on the Timed Observation Theory, the available data can be used to build the model (cf. the scenario model step) but also to evaluate the coherence of the resulting models as illustrated in the preceding section.

- Symbol Driven.
  It can be quite surprising that a Knowledge Engineering methodology claims to be *symbol driven*: the aim is to avoid the introduction of erroneous interpretation in the models. This is particularly important when modeling a dynamic process because precisely, the description of its dynamics is not natural and so very difficult for both the experts and the Knowledge Engineers. The didactic example of [CPR00] clearly illustrates this point: it is striking to notice the difference between the scenario and the generic models.

- Granularity.
  The definition of a dynamic process as a *network of dynamic process* avoid the necessity of describing complex behavior in a unique behavioral model, most oftently unfeasible as the didactic example shows.

The TOM4E (Timed Observation Management for Explanation) algorithm aims at diagnosing a network of dynamic processes according to the models made with the TOM4D methodology. This algorithm has been implemented with a generator of *abstract binary observers networks* (cf. [LG06]) that transforms a set of timed binary relation of an observation model into a network of *abstract binary observers* that produces the same result as TOM4E. The *Detect-Describe-Explain* main functions of TOM4E are distributed on each individual *abstract binary observers* corresponding to each sub-processes of a network.

The advantage is the *abstract binary observers* technology is that such a network realizes a purely *time driven* and *distributed* computation. As a consequence, the computation of the TOM4E algorithm is very fast and parsimonious in memory. This means that is can easily be used in real time, even with very large scale networks of dynamic processes. The TOM4E algorithm being *Any Time*, with the *abstract binary observers* technology it can also be used *Real Time*.

The TOM4D methodology and the TOM4E algorithm have been applied on a real world application, the French Sapin's hydraulic dam deterioration [PRB06]. Dams are heterogeneous structures featured by complex behaviors that evolve through time because of their natural aging. This aging can be accelerated by environmental causes (climatic conditions, floods and earthquakes) or by internal causes (poor design or construction conditions, insufficient or inadequate maintenance...). These causes involve, during the life of the structure, the occurrence and the development of deterioration phenomena, more or less dependent and stemming from miscellaneous and complex sources.

The example of Sapin's dam makes the demonstration of the interest of the TOM4E algorithm coupled with TOM4D modeling: using them, the first Sapin's dam problem would have been identified eight years before its quasi-failure, and the presence of water is highlighted seven years before. This illustrates that TOM4D models and the TOM4E algorithm are then operational tools to diagnose real world dynamic processes.

7.3 Perspectives

The automation of the diagnosis approach based on the TOM4E algorithm is currently under development in Java. It is expected then that this implementation can be used in the hydraulic dam Project for detecting the automatic identification of: (i) the phenomena involved in the deterioration of dam reliability and safety; and (ii) the causes of these deterioration. This work is interesting as, currently, in different countries of the world dam diagnoses are performed by expert engineers who make proposals for corrective actions during dam reviews on the basis of data collected from the structure. Numerous communications of the triennial congress organized by the International Committee on Large Dams have dealt with the development of expert-based approaches for dam diagnosis and several works dealing with. An application on more complex systems, in the civil engineering domain or other domains, is also important to improve and to extend the proposed diagnosis approach.

On the other hand, one of the main next step of our research concerns the use of the results obtained to carry out a prognosis by designing an algorithm dedicated to the automatic prediction of system failure of the system before their occurrences and than to avoid their occurrence. Failure prognosis is an active area of research. In [VLR^{+}06] for example, the failure prognosis is issued when a failure can occur with a very high probably. The introduction of a stochastic approach in the diagnosis algorithm could then be an interesting extension of our approach, making room for a link between TOM4D and TOM4L models.

Besides the TOM4D Knowledge Engineering methodology presented in this document, the mains applications of the TOT is a Knowledge Discovery in Database process called TOM4L (Timed Observation Mining for Learning, [LGBG05, BLG10, LGA12]) and an operational method of abstraction called TOM4A (Timed Observation Method for Abstraction, [Pom12a, PMAP12]).

The main advantages of these applications is that, being based on the same mathematical theory, TOM4D, TOML and TOM4A are compatible together so that their results can easily be merged in order to build and to validate models from the available real-world timed data.

Bibliography

[AD94] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[AF97] G. E. Alcorta and P. Frank. Deterministic nonlinear observer-based approaches to fault diagnosis : A survey. *Control Engineering Practice*, 5(5):663–670, 1997.

[Ahb10] A. Ahbad. *Contribution to Bayesian Networks Learning from Timed Data to Diagnose Continuous Dynamic Process.* PhD thesis, Aix-Marseille University, Marseille (France), June 2010.

[AL01] M Alavi and DE Leidner. Knowledge management and knowledge management systems. *Conceptual Foundations and Research Issues.*, 2001.

[ALG10] A. Ahdab and M. Le Goc. Learning Dynamic Bayesian Networks with the TOM4L Process. In *Proceedings of the Fifth International Conference on Software and Data Technologies (ICSoft 2010)*, volume 2, pages 353–363, 2010.

[Ang10] C. Angeli. Diagnostic expert systems - from expert's knowledge to real-time systems. In Sajja and Akerkar, editors, *Advanced Knowledge Based Systems: Model, Applications and Research*, volume 1, pages 50–73. 2010.

[BCTTD98] V. Brusoni, L. Console, P. Terenzioni, and D. Theseider-Drupré. A spectrum of definitions for temporal model-based diagnosis. *Artificial Intelligence*, 102(1):39–79, 1998.

[BdV94] J. Breuker and W. Van de Velde. Commonkads library for expertise modelling. *IOS Press*, 1994.

[Bea71] R. V. Beard. *Failure accommodation in linear systems through self reorganization.* PhD thesis, Massachusetts Institute of Technology (MIT), 1971.

[Ben10] N. Benayadi. *Contribution à la découverte de connaissances à partir de données datées.* PhD thesis, Univertité Paul Cezanne Aix-Marseille III, 2010.

[Bis94] C. M. Bischop. Neural networks and their applications. *Revue science instrument*, 65(6):1803–1832, 1994.

[BLG10] N. Benayadi and M. Le Goc. Mining Timed Sequences with TOM4L Framework. In *Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS 2010)*, volume 2, pages 111–120, 2010.

[Bou05] P. Bouché. *Une approche stochastique de modélisation de séquences d'événements discrets pour le diagnostic des systèmes dynamiques.* PhD thesis, Aix-Marseille University, 2005.

[BSL01] B. Ould Bouamama, M. Staroswiecki, and R. Litwak. *Automatique et statistiques pour le diagnostic*, chapter Surveillance d'un générateur de vapeur, pages 166–199. B. Dubuisson, paris, france edition, December 2001.

[CD99] L. Consoleánd O. Dressier. Model-based diagnosis in the real world: lessons learned and challenges remaining. In *16th International Joint Conference on Artificail Intelligence (IJCAI'99)*, volume 2, pages 1393–1400, 1999.

[CD00] M.-O. Cordier and C. Dousson. Alarm drive, monitoring based on chronicles. In *4th Symposium on Fault Detection, Supervision and Safety for Technical (Safeprocess)*, pages 286–291, Budapest (Hongrie), June 2000.

[CDD$^+$00a] M.-O. Cordier, P. Dague, M. Dumas, F. Lev́y, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès. Ai and automatic control approaches of model-based diagnosis : Links and underlying hypotheses. 2000.

[CDD$^+$00b] M.-O. Cordier, P. Dague, M. Dumas, F. Lev́y, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès. A comparative analysis of ai and control theory approaches to model-based diagnosis. In *14th European Conference on Artificial Intelligence*, Berlin, Germany, 2000.

[CDL$^+$04] M.-O. Cordier, P. Dague, F. Lev́y, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès. Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man and Cybernetic*, 34(5):2163–2177, October 2004.

[CDPTM13] M.-O. Cordier, P. Dague, Y. Pencolé, and L. Travé-Massuyès. *Diagnostic et supervision : approches à base de modèles.* 2013.

[CGTT93] L. Chittaro, L. Guida, L. Tasso, and E. Toppano. Functional and teological knowledge in the multi-modeling approach for reasoning about physical systems: A case study in diagnosis. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 23, pages 1718–1751, 1993.

[CIG94] CIGB. Bulletin 93-ageing of dams and appurtenant works. *review and recommendations. CIGB, Paris.*, 1994.

[CIG95] CIGB. Ruptures de barrage - analyse statistique. *Bulletin de la Commission Internationale des Grands Barrages*, 99, 1995.

[CL99] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems.* Kluwer Academic Publishers, 1999.

[CLGR$^+$07] M.-O. Cordier, X. Le Guillou, S. Robin, L. Rozé, and T. Vidal. Distributed chronicles for online diagnosis of web services. In *18th International Workshop on Principles of Diagnosis (DX'07)*, Nashville, Tennessee (USA), May 2007.

[Con00] L. Console. Model-based diagnosis history and state of the art., 2000.

[CPB10] C. Curt, L. Peyras, and D. Boissier. A knowledge formalization and aggregation-based method for the assessment of dam performance. *Computer-aided Civil and Infrastructure Engineering*, 2010.

[CPR00] L. Console, C. Picardi, and M. Ribaudo. Diagnosing and diagnosability analysis using pepa. In *14th European Conference on Artificial Intelligence (ECAI'2000)*, pages 131–135, Berlin, Germany, August 2000.

[CPR02] L. Console, C. Picardi, and M. Ribaudo. Process algebras for system diagnosis,. *Artificial Intelligence Journal*, 142(1):19–51, 2002.

[CR99] L. Chittaro and R. Ranon. Diagnosis of multiple faults with flow-based functional models: the functional diagnosis with efforts and flows approach. *Reliability Engineering and System Safety, 64(2):137-150, 1999.*, 1999.

[CR04] L. Chittaro and R. Ranon. Hierarchical model-based diagnosis based on structural abstraction. *Advanced Engineering Informatics*, 2004.

[CT90] L. Console and P. Torasso. Integrating models of the correct behavior into abductive diagnosis. In *9th European Conference on Artificial Intelligence (ECAI'90)*, number 160-166, Stockholm, Sueden, 1990.

[CT91] L. Consoleánd P. Torasso. A spectrum of logical definitions of model-based diagnosis. *Computational Intelligence*, 7(3):133–141, 1991.

[Cur13] C. Curt. Dynamic theory of organizational knowledge creation. *In Dams: Structure, Performance and Safety Management*, 2013.

[CW84] E. Y. Chow and A. S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transaction on Automatic Control*, 29:603–614, 1984.

[Dag01] P. Dagues. *Diagnostic, Intelligence Artificielle et Reconnaissance des Formes*, chapter Théorie logique du diagnostic à base de modèles, pages 11–104. 2001.

[Dam99] A. Damasio. The feeling of what happens: Body and emotion in the making of consciousness, harcourt. 1999.

[Dam05] A. Damasio. Descartes' error: Emotion, reason, and the human brains. 2005.

[DH92] R. Davis and W. Hamscher. *Model-Based Reasoning: Troubleshooting.* 1992.

[DK03] J. De Kleer. Fundamentals of model-based diagnosis. In *14th International Workshop on Principles of Diagnosis (DX'03)*, 2003.

[DKMR92] J. De Kleer, R. Mackworth, and R. Reiter. Characterizing diagnoses and systems artificial intelligence. *Artificial Intelligence*, 56(2-3):197–222, 1992.

[DKW87] J. De Kleer and G. Williams. Diagnosing multiple faults. *Artificial Intelligence, 32:97-130.*, 1987.

[DKW89] J. De Kleer and G. Williams. Diagnosis with behavioral modes. *11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1324–1330, 1989.

[Dou94] C. Dousson. *Suivi d'évolutions et reconnaissance de chroniques.* PhD thesis, Université Paul Sabatier, Toulouse (France), 1994.

[Dub01] Bernard Dubuisson, editor. *Diagnostic, intelligence artificielle et reconnaissance des formes.* Traité IC2 Information - Commande - Communication. Hermes Science Publications, 8, quai du Marché-Neuf, 75004 Paris, Juin 2001.

[DYHAA06] H. Derbel, N. Yeddes, B. Hadj-Alouane, and H. Alla. Diagnosis of a class of timed discrete event sytems. In *8th International Workshop on Discrete Event Systems (WODES'06)*, pages 256–261, Michigan (USA), 2006.

[EMG98] A. Evsukoff, J. Montmain, and S. Gentil. Causal model based supervising and training. In *IFAC Workshop on On-line Fault Detection and supervision in the Chemical Process Industrie*, Solaize, Lyon, France, 1998.

[FD00] P. M. Frank and S. X. Ding. Current developments in the theory of fdi. In *SAFEPROCESS 2000*, pages 16–27, Budapest, Hungary, 2000.

[FFS00] M. Foster, R. Fell, and M. Spannagle. A method for assessing the relative likelihood of failure of embankment dams by piping. *Canadian Geotechnical Journal, 37, 1025-1061.*, 2000.

[Flo08] L. Floridi. The method of levels of abstraction. *Minds and Machines*, 18:303–329, September 2008.

[Fra96] P. M. Frank. Analytical and qualitative model-based fault diagnosis – a survey and some new results. *European Journal of Contro*, 2:6–28, 1996.

[GCL05] A. Grastien, M. Cordier, and C. Largouet. Premiers pas vers le diagnostic incrémental de systèmes à événements discrets. *Revue d'intelligence artificielle*, 2005.

[GD04] B. Guerraz and C. Dousson. Chronicles construction starting from the fault model of the system to diagnose. In *15th International Workshop on Principles of Diagnosis (DX'04)*, pages 51–56, Carcassone (France), June 2004.

[Ger91] J. Gertler. Analytical redundancy methods in fault detection and isolation – survey and synthesis. In *IFAC Symposium on Fault Detection Supervision and Safety for Technical Process*, pages 9–22, Baden Baden, Germany, 1991.

[Ger97] J. Gertler. Fault detection and isolation using parity relations. *Control Engineering Practice*, 5(5):653–661, 1997.

[Ger98] J. Gertler. *Fault Detection and diagnosis in engineering systems.* Marcel Dekker Inc., 1998.

[GL03] S. Genc and S. Lafortune. Distributed diagnosis of discrete-event systems using petri nets. In *Applications and Theory of Petri Nets 2003, 24th International Conference ICATPN*, Eindhoven, The Nederlands, June 2003.

[GS90] J. Gertler and D. Singer. A new structural framework for parity equation-based failure detection and islolation. *Automatica*, 26(2):381–38, 1990.

[HP00] L. Hollowat and N. Pandalai. Template languages for fault monitoring of timed discrete event processes. *IEEE Transactions on Automatic Control*, 45(5):868–882, 2000.

[IB97] R. Isermann and P. Balle. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719, 1997.

[Jon73] H. L. Jones. *Failure detection in linear systems.* PhD thesis, Massachusetts Institute of Technology (MIT), 1973.

[KN02] M. Krysander and M. Nyberg. Structural analysis utilizing mss sets with application to a paper plant. In *International Workshop on Principles of Diagnosis (DX'02)*, 2002.

[LDM09] Y. Li, P. Dague, and T. Melliti. A decentralized model-based diagnosis for bpel services. In *ICTAI09*, pages 609–616, 2009.

[LG04] M. Le Goc. SACHEM, a real-time intelligent diagnosis system based on the discrete event paradigm. *Simulation*, 80(11):591–617, 2004.

[LG06] M. Le Goc. *Notion d'observation pour le diagnostic des processus dynamiques : Application à Sachem et à la découvertes de connaissances temporelles.* PhD thesis, 2006.

[LGA12] M. Le Goc and A. Ahdab. *Learning Bayesian Networks From Timed Observations.* LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012.

[LGBG05] M. Le Goc, P. Bouché, and N Giambiasi. Stochastic modeling of continuous time discrete event sequence for diagnosis. In *16th International Workshop on Principles of Diagnosis (DX'05)*, 2005.

[LGM07] M. Le Goc and E. Masse. Towards a multi-modeling approach of dynamic systems for diagnosis. In *2nd International Conference on Software and Data Technologies, ICSoft*, pages 22–25, Barcelona, Spain, July 2007.

[LGMC08] M. Le Goc, E. Masse, and C. Curt. Modeling processes from timed observations. In *3rd International Conference on Software and Data Technologies, ICSoft*, Porto, Portugal, July 2008.

[LY77] H. E. Lambert and G. Yadigaroglu. Fault trees for diagnosis of system condition. *Nuc. Sci. and Eng.*, pages 20–34, 1977.

[Mer74] P. M. Merlin. *A study of the recoverability of computing systems.* PhD thesis, Department of Information and Computer Science, University of California, Irvine, Califormia (USA), 1974.

[MK92] USA Morgan Kaufmann, San Mateo, editor. *Readings in Model-Based Diagnosis.* W. Hamscher and L. Consoleánd J. De Kleer, 1992.

[ML07] E. Masse and M. Le Goc. Modeling dynamic systems from their behavior for a multi model based diagnosis. In *8th International Workshop on the Principles of Diagnosis (DX'07)*, pages 322–329, May 2007.

[NK98] I. Nonaka and N. Konno. The concept of "ba": Building a fundation for knowledge creation. *California Management Review*, 40(3):40–54, 1998.

[Non91] I. Nonaka. The knowledge-creating company. 1991.

[Non94] I. Nonaka. Combining knowledge-based method and possibility theory for assessing dam performance. 1994.

[NP90] K. S. Narenda and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, 1990.

[Nyb01] M. Nyberg. A general framework for fault diagnosis based on statistical hypothesis testing. In *12th International Workshop on Principles of Diagnosis (DX'01)*, Working Notes, Sansicario, Italy, 2001.

[Pat97] R. J. Patton. Fault tolerant control: the 1997 situation. In *IFAC Symposium on Fault Detection Supervision and Safety for technical process (IFAC'97)*, volume 2, pages 1033–1055, Hull, United Kingdom, August 1997.

[Pau86] L. P. Pau. Survey of expert systems for fault detection, test generation and maintenance. *Expert Systems*, 3:100–11, 1986.

[PBR04] L. Peyras, D. Boissier, and P. Royet. Approches de l'analyse de risques en geńie civil. *Revue Française de Geńie Civil*, 8:931–952, 2004.

[PBRV04] L. Peyras, D. Boissier, P. Royet, and A. Vergne. Diagnostic et analyse de risques liés au vieillissement des barrages : développement de méthodes d'aide à l'expertise. *Ingeńieries*, 38:3–12, Juin 2004.

[PC91] R. J. Patton and J. Chen. A review of parity space approaches to fault diagnosis. In *IFAC Safeprocess symposium*, Baden-Baden, Germany, 1991.

[PCR02] Y. Pencolé, M.-O. Cordier, and L. Rozé. A decentralized model-based diagnostic toll for complex systems. *International Journal on Artificial Intelligence Tools*, 11(3):327–346, 2002.

[PL76] G. J. Powers and S. A. Lapp. Computer-aided fault tree synthesis. *CEP*, 12(4):89–93, April 1976.

[PL10] L. Pomponio and M. Le Goc. Timed observations modelling for diagnosis methodology: a case study. In M. Virvou In J. A. Moinhos Cordeiro and editors B. Shishkov, editors, *5th International Conference on Software and Data Technologies (ICSoft 2010)*, volume 2, pages 504–507, Athens (Greece), July 2010. SciTePress.

[PLG14] L. Pomponio and M. Le Goc. Reducing the gap between experts' knowledge and data: The tom4d methodology. *Data & Knowledge Engineering*, DOI 10.1016/j.datak.2014.07.006, July 2014.

[PMAP12] L. Pomponio, Le Goc M., A. Anfosso, and E. Pascual. Levels of abstraction for behavior modeling in the gerhome project. *International Journal of E-Health and Medical Communications*, 3(3):12–28, September 2012.

[Pol66] M. Polanyi. The tacit dimensione. 1966.

[Pom12a] L. Pomponio. *Definition of a Human-Machine Learning Process from Timed Observations: Application to the Modelling of Human Behaviour for the Detection of Abnormal Behaviour of Old People at Home.* PhD thesis, Aix-Marseille University, Marseille (France), June 2012.

[Pom12b] L. Pomponio. *Definition of a Human-Machine Learning Process from Timed Observations: Application to the Modelling of Human Behaviour for the Detection of Abnormal Behaviour of Old People at Home.* PhD thesis, Aix-Marseille University, Marseille (France), June 2012.

[PR01] M. Poupard and P. Royet. La surveillance des barrages. in proceedings of cfgb colloque technique. *Aix-en-Provence, France.*, 2001.

[PRB06] L. Peyras, P. Royet, and D. Boissier. Dam ageing diagnosis and risk analysis: Development of methods to support expert judgment. *NRC Canada*, 2006.

[Ram74] C. Ramchandani. Analysis of asynchronous concurrent systems by timed petri nets. Project mac-tr 120, Massachusetts Institute of Technology (MIT), February 1974.

[Rei87] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence, 32(1), 57-96*, 1987.

[RK83] R. Rosenberg and D. Karnopp. Introduction to physical system dynamics. In *McGraw-Hill, Inc. New York, NY, USA.*, 1983.

[SAA$^+$00] Th. Schreiber, J. M. Akkermans, A. A. Anjewierden, R. de Hoog, N. R. Shadbolt, W. Van de Velde, and B. J. Wielinga. Publication, knowledge engineering and management the commonkads methodology. *MIT Press*, 2000.

[SBF98] R Studer, VR Benjamins, and D Fensel. Knowledge engineering: Principles and methods. 1998.

[SD89] P. Struss and O. Dressler. Physical negation : Integrating fault models into the general diagnostic engine. *11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1318–1323, 1989.

[Sha84] C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1984.

[SSD97] P. Struss, M. Sachenbacher, and F. Dummert. Diagnosing a dynamic system with (almost) no observations, a case suty in off-board diagnosis of the hydraulic circuit of an anti-lock bracking system. In *Working notes of the 11th International Workshop on Qualitative Reasoning (QR'97)*, Cortona, Italy, 1997.

[SSL$^+$95] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.

[SSL$^+$96] M. Sampath, R. Sengupta Sengupta, S. Lafortune Lafortune, K. Sinnamohideen, and D. Teneketzis. Failure diagnosis using discrete-event models. 1996.

[Str91] P. Struss. What's in sd? towards a theory of modeling for diagnosis. In *International Workshop on the Principles of Diagnosis (DX'91)*, pages 41–51, Milan, Italy, 1991.

[SW89] W. Scherer and C. White. A survey of expert systems for equipment maintenance and diagnostics. In S. Plenum Publishing Inc., editor, *Knowledge-based system Diagnosis, Supervision, and Control.* Tzafestas, 1989.

[TBP09] A. Talon, D. Boissier, and L. Peyras. Analyse de risques, identification et estimation, démarches d'analyse de risques, méthodes qualitatives d'analyse de risque. Université Numérique Ingénierie et Technologie (H. Niandiou), 2009.

[TCG90] A. K. A. Toguyeni, E. Craye, and J. C. Gentina. A method of temporal analysis to perform online diagnosis in the context of flexible manufacturing system. In *16th Annual conference of IEEE Industrial Electronics Society (IECON'90),*, volume 1, pages 445 –450, Pacific Grove, Californie (USA), November 1990.

[TMEPTS01] Louise Travé-Massuyès, Teresa Escobet, Renaud Pons, and Sebastian Tornil-Sin. The ca∼en diagnosis system and its automatic modelling method. *Computación y Sistemas*, 5(2), 2001.

[TP79] T.L. Teague and G.J. Powers. *Diagnosis Procedures from Fault Trees.* DRC. Design Research Center, Carnegie-Mellon University, 1979.

[Tri02] S. Tripakis. Fault diagnosis for timed automata. In *Formal Techniques in Real Time and Fault Tolerant Systems (FTRTFT'02)*, volume 2469 of *LNCS*. Spinger, 2002.

[VLR$^+$06] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu. Intelligent fault diagnosis and prognosis for engineering systems. *Hoboken, NJ : John Wiley and Sons*, 2006.

[Zad65] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338 –353, 1965.

[Zan04] C. Zanni. *Proposition of a Conceptual Framework for the Analysis, Classification and Choice of Knowledge Based Diagnosis Systems.* PhD thesis, Université de Droit, dÉconocmie et des Sciences dÁix-Marseille, June 2004.

[ZGF06] C. Zanni, M. Le Goc, and C. Frydman. A conceptual framework for the analysis, classification and choice of knowledge-based diagnosis systems. *KES - International Journal of Knowlegde-Based and Intelligent Engineering Systems, Kluwer Academic Publishers*, 10(2):113–138, 2006.

Résumé

Cette thèse concerne le diagnostic de processus dynamiques basé sur la Théorie des Observations Datées, une théorie mathématique conçue pour la modélisation et le raisonnement à partir de données datées. Les contributions présentées dans ce mémoire sont 1) une extension de la méthodologie d'ingéniérie des connaissances TOM4D (Timed Observation Modeling for Diagnosis) aux réseaux de processus dynamiques, 2) l'algorithme temps réel et any-time TOM4E (Timed Observation Management for Explanation) qui utilise les modèles TOM4D pour diagnostiquer les comportements dans un réseau de processus dynamiques à partir de données datées et 3) l'application de TOM4D et TOM4E au diagnostic du barrage hydraulique des Sapins (France), un problème particulièrement difficile. TOM4D est une approche de diagnostic à partir de multiples modèles dirigée par la syntaxe où l'introduction de la sémantique est contrôlée par la combinaison de l'approche conceptuelle de CommonKADS au tétraèdre des états de la physique newtonienne. Les fonctions Detect, Describe et Explain de TOM4E utilisent les modèles d'observation déduit des modèles de comportement de TOM4D pour identifier les comportements potentiels des processus. Pour des raisons de simplicité, la présentation de TOM4D et de TOM4E est effectuée à l'aide d'un exemple didactique tiré de la littérature spécialisée dans le domaine du diagnostic. L'application au diagnostic du barrage des Sapins démontre l'intérêt de l'approche : leur usage aurait permit d'identifier le premier problème huit ans avant sa quasi-destruction, la présence d'eau étant mise en évidence sept ans avant.

Mots clés : Diagnostic à base de Modèles Multiples, Modélisation de réseau de processus dynamiques, Ingéniérie des connaissances, Barrage hydraulique

Abstract

This thesis proposes a diagnosis approach of dynamic process based on the Timed Observation Theory, a mathematical framework for modeling and reasoning about dynamic process from timed data. The contributions of this works are i) an extension of the TOM4D (Timed Observation Modeling for Diagnosis) Knowledge Engineering methodology to networks of dynamic processes, ii) a real-time and any-time diagnosis algorithm called TOM4E (Timed Observation Management for Explanation) that uses the TOM4D models to diagnose behaviors in a network of dynamic processes and iii) the application of TOM4D and TOM4E to the diagnosis of the French Sapin's hydraulic dam, a particularly difficult real-world diagnosis problem. TOM4D is a is a primarily syntax-driven approach of Multi-Model Based Diagnosis where semantic content is introduced in a gradual and controlled way through the combination of the CommonKADS conceptual approach and the Tetrahedron of States of Newton's physical laws. TOM4E algorithm is based on the Dectect, Describe and Explain functions which uses observation models translated from the TOM4D behavioral models. For simplicity reasons, the presentation of TOM4D and TOM4E is made with a unique didactic example provided from the literature of the diagnosis domain. The example of Sapin's dam makes the demonstration of the interest of the proposed approach: using them, the first Sapin's dam problem would have been identified eight years before its quasi-failure, and the presence of water being highlighted seven years before.

Keywords: Multi Model Based Diagnosis, Network of Dynamic Process Modeling, Knowledge Engineering, Hydraulic Dam

LSIS, Laboratoire des Sciences de l'Information et des Systèmes, UMR CNRS 7296
Falculté de Saint Jérôme, Avenue Escadrille Normandie Niemen
1397 Marseille Cedex 20