Monitoring and alarm interpretation in industrial environments

S. Cauvin, IFP, 1-4 Avenue du Bois Préau, F-92852 Rueil Malmaison cedex
M-O. Cordier, IRISA, Campus de Beaulieu, F-35000 Rennes
C. Dousson, France Telecom – CNET, 2 Avenue Pierre Marzin, F-22307 Lannion cedex
P. Laborie, EDF/DER/ER, 1 Avenue du Général de Gaulle, F-92141 Clamart cedex
F. Lévy, LIPN, Avenue J.B. Clément, F-93430 Villetaneuse
J. Montmain, CEA Marcoule, BP 171, F-30205 Bagnols sur Ceze
M. Porcheron, EDF/DER/SDM, 6 Quai Wattier, F-78400 Chatou
I. Servet, LAAS-CNRS, 7 Avenue du Colonel-Roche, F-31077 Toulouse cedex
L. Travé-Massuyès, LAAS-CNRS, 7 Avenue du Colonel-Roche, F-31077 Toulouse cedex

The “ALARM” research group has been running two years as a multi-labs group within the French National Program on Artificial Intelligence PRC-IA. The group’s objective was to bring together representatives of the academic and industrial worlds in order to analyze, in real applications, the various problems raised by alarm interpretation and to define the potential benefits of AI techniques in this field. This paper presents the conclusions stemming from the analysis and discussions which took place during this period. It describes the industrial applications which the members of the group dealt with and compares them in a table with respect to several criteria identified as the most significant.

1. Introduction

The “ALARM” research group focused on the study of tools and methods for building systems aimed at interpreting alarms. This theme is undoubtedly significant from a practical point of view as interpreting alarms is critical to the operation of many industrial sites.

In this paper, the conclusions stemming from the analysis and discussions which took place during this period are presented. This paper does not compare AI techniques with other possible techniques such as those proposed by the process control, statistics or signal processing communities. On the other hand, it is primarily concerned with monitoring and diagnosis. Problems related to prediction and decision-making are not addressed.

In section 2, we state the role of alarm interpretation in complex process monitoring. In section 3, we define what “alarm interpretation” means. In section 4, we present the techniques used in the industrial applications which the members of the group dealt with. In section 5, we describe each application in detail, in a common framework in order to highlight common features. The last section compares these applications in a table with respect to several criteria identified as the most significant.

2. Alarms and monitoring

Any physical system evolves with time, either due to its own dynamics or under the impact of external actions or events. Informally, monitoring a dynamic system can be viewed as performed by a high level module which keeps track of the system continuously, analyses all situations encountered, communicates with human operators and suggests decisions to be taken in case of dysfunctions. Monitoring heavily relies on alarms, which rely in turn on sensor values. Such systems are encountered in telecommunication and power distribution networks, as well as in major industrial plants such as nuclear plants, refining or petrochemical plants, etc. They have certain common features. Numerous alarms which indicate either dysfunctions or other significant events are continuously produced. The monitoring system can receive up to several hundred messages per second, making interpretation difficult. The task is further complicated by other features of these systems:
they are dynamic: the behavior of individual components evolves over time and, in some cases, changes are due to the activity of other components;

alarms are not necessarily received in the same order as the one in which they were emitted. Propagation times and, sometimes, transmission paths must be taken into account, both to reconstruct the order of events and to decide when all relevant messages have been received;

alarms are not independent: some are merely the consequence of others;

some alarms may be lost or masked: they are emitted, but due to the dysfunctions of an intermediate element, they never reach the monitoring system.

Therefore, even the absence of an alarm can provide helpful information about the state of the process.

There is very little consensus as to the architecture of the ideal monitoring system, partly because of the diversity of the processes that must be monitored (continuous or discrete for instance). In general, it consists in several cooperating modules (see figure 1):

- a detection module which gathers elementary information provided by the sensors and decides whether the evolution of the process is normal;
- a diagnosis module, responsible for one or more of the following tasks:
  * identifying characteristic situations (especially abnormal situations);
  * localizing the faulty components responsible for the situation;
  * determining the primary causes of the abnormalities detected;
- a decision module which determines the actions which can be undertaken to reach the objective or bring back the process in normal conditions; these actions are generally presented to the operators.

These modules rely on a knowledge base which contains:

- a library of more or less detailed behavioural models of the physical system, under normal operating conditions and, when possible, in degraded modes or in the presence of failures,
- the objectives to be met by the physical system, possibly together with plans of actions, in normal or abnormal conditions,
- a description (or several descriptions on various levels) of the present state of the system, summarizing recent observations or expressing these in a language which can be understood by the operator.

Fig. 1. Typical architecture of a monitoring system

3. Alarm interpretation

3.1. What is an alarm?

The very concept of “alarm” varies from one application to another, which can complicate the understanding and expression of the results. This section gives the definition that has been gradually adopted by the members of the “ALARM” group after analyzing together several industrial applications in various sectors (nuclear power systems, telecommunication networks, petroleum industry, etc.). This definition is based on the notion of “event” whose definition is given first, followed by the one of an alarm:

“An event is a piece of information extracted from continuous or discrete signals emitted by a component (significant variations in a variable message emitted by the system) or data about the context (repair, actions, observations related to the environment, etc.). It is dated and instantaneous.”

“An alarm is a discrete indicator emitted by the monitoring system on the basis of events; it is intended to trigger a human or automated reaction.”

In a certain number of cases, there are several levels of alarms: safety alarms which trigger automated reactions and process alarms intended to draw attention so that preventive or corrective actions are taken. The triggering of an alarm can be determined:

- either directly by a physical mechanism following simple event detection (such as an alarm indicating a pump failure when no output flow is detected);
- or by calculation; for example when a signal is compared to a reference.
This rises the following problems:

- what should the reference be?
- what information in the signal should be compared to the reference; should the signal be processed; should one or several values be compared?
- what means of comparison should be adopted?
- such triggering of alarms can be determined at various levels in the overall data-processing procedure; it may be at the level of the sensors, that of the central system, or at an intermediate level.

Under real operating conditions, the main problems encountered in interpreting alarms are:

- the absence of emission of an alarm, the loss of an alarm during transmission, or the masking of one or several alarms by others;
- cascade phenomena related to the strong interdependence among system variables;
- the pertinence and degree of discrimination of the sets of alarms obtained (do they really enable detection of the real problems?);
- the importance of the context.

These different problems, which are closely related to the use one makes of alarms, are discussed in the following sections.

3.2. Generation of alarms

Alarms may be used for two different purposes, which do not imply the same time constraints.

When they are intended for the operator, they are processed on-line. The purpose of monitoring is thus supervisory control and therefore the interpretation must be done in real-time. The operator has a short-term optimization objective: in general, the goal is to remain as close as possible to ideal operating conditions, allowing for the variability of inputs and the natural evolution of the processes. For instance, the structural shifts in the system (wear on parts, slow modifications in the properties of the components, etc.) are not taken into account as such, and they are corrected by adjusting the control parameters.

When they are intended for the maintenance expert, processing may be differed. Off-line analysis is more detailed, since the objective of maintenance is to foresee incidents and to schedule maintenance operations so as to limit failures and interruptions in service to a minimum. Background on the system, slow evolution of its characteristics and the recurrence of phenomena corrected by the process must be interpreted in order to recognize those structural changes which need to be dealt with.

Diverse objectives imply diverse processing methods as well. When the operator controls the final decisions to be undertaken an important issue is to avoid the “cognitive overload” problem characterized by a flood of data, most of which is redundant. The aim is hence to “intelligently” organize the information provided to the operator. When it is possible that decisions are taken with no human intervention and when it is important to react quickly, interpretation of alarms may as well aim at automatically triggering the reactions appropriate to the evolution observed on the system.

3.3. Prioritizing alarms

In the cases when the operator retains the initiative for undertaking actions, the objective is merely to facilitate his understanding of the data. Alarms are often redundant, either because several sensors serve to detect a single phenomenon, or because the cause phenomenon triggers a cascade of secondary anomalies which are not significant on their own, and whose display merely burdens the operator. The aim is therefore to assist the operator in filtering the alarms so as to display only the most significant ones.

Several methods may be used for filtering:

- one can combine measurements provided by elementary sensors to obtain a more synthesized display, for example a graph can be analyzed and presented in a symbolic form;
- one can use a causal network expressing failure modes and their interdependency, or more directly the dependency among alarms, and display only those which appear the farthest upstream (or corresponding to the failure modes farthest upstream) in the network;
- one can also use simulation to verify whether or not the consequences of a hypothetical fault coincide with the real measurements, and in this case, filter out the redundant alarms.

Another means of helping the operator is by classifying alarms. They are pre-analyzed and presented differently, according to their importance and degree of emergency. Monitoring systems often distinguish several levels of alarms:

- the highest level indicates a direct threat to safety and triggers shutdown of all or part of the supervised system;
– warnings require later repair measures, but generally not immediate shutdown;
– on the lowest level, alarms are merely process indicators used to maintain the system as close as possible to the optimum behavior.

3.4. Processing alarms

Automated interpretation of alarms may have one of several objectives: either to determine the causes of a dysfunction and provide explanations to the operator, or to predict future behaviour of the system so as to assess the degree of emergency of a situation, or to react automatically by triggering physical mechanisms.

Determination of causes is similar to the overall process of diagnosis, and therefore calls for the various techniques in this field. Among the possible causes, we can distinguish internal faults related to equipment failures and external faults corresponding to inputs that are outside the acceptable range. It may be necessary to provide explanations, and therefore to describe explicitly some underlying implicit causal relations. For this purpose, causal networks are again a useful way to work back to the explanations. It must nonetheless be noted that their use for this purpose is different than their use for filtering alarms: for explanation purposes, neglecting one causal relation can render recognition of the primary cause impossible, whereas for alarm filtering, it merely leads to useless redundancies.

Prediction can use more or less complete, quantitative or qualitative models. Such models are rarely available and tractable for the whole system.

Some alarms or alarm configurations are associated with an immediate reaction. Whenever the most crucial factor is the speed of reaction, this association is pre-calculated and managed by automated control mechanisms. In other situations, there is less emergency (and the reaction can be within minutes or even hours) or less potential seriousness. It is therefore beneficial to have access to cheaper techniques for recognizing significant situations, and to pursue research on the various means for automated patterns recognition.

4. AI techniques and tools for alarm management

4.1. Causal graphs relating failures and manifestations

The idea of exploiting knowledge of causes for the purpose of diagnosis is quite natural. A “dysfunction” can be described in a simple way by relations associating its original causes (component failures, illness, etc.) to observable manifestations, or symptoms. Having a theory which models this kind of relations, the diagnostic problem consists in using the theory to seek satisfactory explanations for the observed symptoms.

The basic inference mechanism of this type of reasoning which “moves from effects to causes” is called abductive. It can be summarized as follows:

Given fact “B” and the association (causal relation):

“A → B” (“A” causes “B”), infer “A is possible”

This is a relatively old approach to diagnosis which was extensively used in medical diagnosis in the early days. Abductive reasoning and its application to diagnosis problems was formalized in the mid-eighties [1]. More recently, these methods received even more attention [2].

Let us look more closely at the formalization of abductive diagnosis as described in [3, 4]. A failure model is considered as a theory constituted by a set of relations between the observations expected when the system is behaving abnormally and the “explanations” one can provide for them. The logical structure of these relations is as follows:

\[ C \rightarrow E \quad : \quad C \text{ causes } E \]
\[ C \land C' \rightarrow E \quad : \quad C \text{ and } C' \text{ cause } E \]
\[ C \land \alpha \rightarrow E \quad : \quad C \text{ can cause } E \text{ under certain unspecified conditions, represented by the abstract condition } \alpha. \]

Certain terms of the failure model are distinguished for the purpose of diagnosis:

– the observable terms, called symptoms, whose truth value can be established by observing the system;
– the primary causes, i.e. the terms which have no cause in the theory; these terms will be the constituents of the elementary explanations produced.

Given a failure model M constituted of a number of relations in the above form and \( \psi \), a set of observations, the positive observations confirm the existence of symptoms, while the negative observations enable negating the existence of others. The explanations of \( \psi \) are the most specific formulas in the language of the primary causes which imply the positive observations and are consistent with the negative observations. Such an explanation is a conjunction of primary causes and of negations of primary causes in M. For the purpose of illustration, let us look at this extract from a failure
model exploited by the DIAPO system for diagnosing the coolant pump sets in EDF nuclear power plants (see section 5.3). The following example is simplified by not considering the time labels attached to the causal relations.

**Shaft vibration** \( \land \alpha \)
- \( \rightarrow \) **Blocking of the pump bearing knee joint**

**Primary water rise** \( \land \beta \)
- \( \rightarrow \) **Breakdown of the pump bearing knee joint**

**Blocking of the pump bearing knee joint**
- \( \rightarrow \) **High D1/D2 ratio**

**Breakdown of the pump bearing knee joint**
- \( \rightarrow \) **High D1/D2 ratio**

**Blocking of seal-1**
- \( \rightarrow \) **Decrease of QFJ1 flow**

**High D1/D2 ratio** and **Decrease of QFJ1 flow** are symptoms. **Blocking of the pump bearing knee joint**, **Breakdown of the pump bearing knee joint** and **Blocking of seal-1** are diagnostic hypotheses; \( \alpha \) and \( \beta \) are abstract conditions.

Given \( \psi = \{ \text{Decrease in QFJ1 flow, High D1/D2 ratio} \} \) the full set of observations, the explanation formula is:

\[
F = (\text{Blocking of seal-1} \land \text{Primary water rise} \land \beta) \lor (\text{Blocking of seal-1} \land \text{Shaft vibration} \land \alpha)
\]

There exists many extensions of this principle, particularly to combine abductive and time-based reasoning. This is especially important in applying these methods to dynamic systems subject to continuous monitoring. The observations are dated, and time spans can be added to the causal relations which model the new temporal relations between the occurrence of the causes and that of their effects. The major difficulty consists in exploiting the two types of knowledge, causal and temporal, together. For example, [5] propose to separate knowledge describing the manifestations caused by the different states of the system indexed by time (behavior models) from that describing the possible transitions from one state to another (transition graphs). The diagnosis is first produced by an abductive resolution in the behavior models, before being validated by verifying its consistency with the possible sequences of change state.

### 4.2. Qualitative models, influence graphs

Qualitative physics aims at representing physical systems, predicting and explaining their behavior on the basis of both the causal, common-sense reasoning used by humans to analyze the environment qualitatively, and the scientific knowledge implicitly used by engineers [6, 7]. The representation of continuous variables must be guided by the following principle: the distinctions made by a quantification must be pertinent to the type of reasoning adopted. Let us add that the models are declarative, in the sense that the representation of the system (the model) and the reasoning supported by the model are independent. Also note that qualitative formalisms are particularly useful in representing imprecise, uncertain or incomplete knowledge.

All these facts explain why qualitative physics have shown very appropriate for supporting the monitoring of continuous processes: with the objective to explain the functioning of the process to the operators, pertinence is indeed more important than precision at the moment of making decisions. Unlike quantitative modeling, the qualitative approach enables one to deal with systems in their operational environment and to understand industrial installations as a whole.

In comparison with pure symbolic models used in logical frameworks, qualitative models have the advantage of accepting arithmetic reasoning. Nevertheless, generally based on intuitive concepts (orders of magnitude, causality, etc.), they enable to derive mechanisms of inference which have analogies with the engineer reasoning about physical systems. In this way, they are well suited to the needs of high-level tasks like monitoring.

Qualitative models are generally used for detection: they provide the reference behavior to which to compare the measurements. The difference between predicted and measured behavior provides the basis for fault detection. As the explicit nature of these models is well adapted to mechanisms of explanation, they can generally be used directly for isolation as well.

Qualitative models can be divided in two classes, according to whether they represent causality explicitly or not.

#### 4.2.1. Models with no explicit causality

In these models, the constraints linking the different variables of the system are of equation type. The basic techniques are constraint resolution and qualitative calculus. The most commonly used qualitative simulator which allows for implementing this type of models is QSIM [8, 9]. The QSIM algorithm recursively examines active states to generate all possible successive states, eliminating those which violate the model constraints in a second step. Because a given state can have several successors, QSIM builds a state tree in
which each branch represents one possible behavior of the physical system.

QSIM has been used as the core of a monitoring system called MIMIC [10]. Two phases can be distinguished in MIMIC: monitoring and diagnosis, both of which call on QSIM. The concepts of monitoring, diagnosis and simulation are associated in a cycle of hypothesis/model building/simulation/comparison. MIMIC uses a library of qualitative models including one model of normal behavior and failure models.

The monitoring phase consists in updating the “monitoring set” composed of those models which are consistent so far with the observations received from the sensors. If M is such a model, the diagnosis phase first identifies all components responsible for deviations between the observations and the present state of M. Then, in a second step, making the hypothesis of a single failure, it finds the variants of M which correspond to the observations by modifying the operating mode of one component. The new models obtained are finally added to the monitoring set.

4.2.2. Models with explicit causality or influence graphs

Davis [11] wrote that a large part of the knowledge needed for analyzing degraded operating conditions builds on understanding the mechanisms in terms of causality. A system can often be described by its structural equations, which are generally algebraic interpretations of the physical laws governing the system; causal ordering provides a guideline for identifying existing links between dependent and independent variables. Indeed a causal structure is a description of the influences that the variables have on each other. The behavior of any system can be at least partially described by an influence graph, i.e. an oriented graph whose nodes represent the variables and arcs, the one-direction relationships among them. This later provides a conceptual tool for examining the way changes in an installation are propagated. Bandekar [12] showed that an explicit representation of relations of this kind is directly useful for diagnosis: in model-based reasoning, this means that knowledge about causal dependencies can be used in searching for the primary deviation in the graph. An influence graph is above all a structure, which can be enriched depending on the knowledge available and the precision required for the diagnosis: the variables can be associated to alarms, but also to more complex features (model-process errors for example); the arcs may indicate only the signs of the influences, but also complex dynamic functions.

Present alarm processing systems provide operators with an unsequenced list of simultaneous alarms which affect the process; later, the operators must interpret and extract the source alarm, which is to say the one that enables explaining all other alarms set off. Influence graph methods aim at helping operators in their isolation task. Given a set of alarms, the influence graph can serve to build a tree whose root is the source alarm and whose branches illustrate the propagation path of alarms. The principle is more or less always the same. The goal is to explain abnormal deviations in the evolution of the variables of an installation with a minimum number of faults at the source. A primary fault is a change in the evolution of a variable which is directly due to a failure or a non-measured degradation, while secondary faults result from the propagation of this deviation in time, causing new deviations. The diagnosis consists in seeking the source variable, whose variation is sufficient to explain all the deviations detected in other variables. The result may be either a degraded arc arriving at the source variable, which corresponds to at least partial lack of one function of the installation (failure of a component), or a degradation which directly affects the source variable. To analyze the propagation path, local tests between one variable and its antecedents are performed, evaluating the consistency between the arcs and the state of the variables; the test is more or less complex, depending on the information carried by the arcs and the definition of the state of a variable.

4.3. Expert systems

Expert systems, like systems based on pattern recognition (see section 4.4), are based on the direct expression of the links between a set of observable events (or symptoms) and a characteristic situation one wishes to identify (in particular, a system dysfunction). Expert systems were the first tools used for diagnosing static systems. They were later extended to diagnosing dynamic systems. Today, they are still the most widely used in industrial monitoring systems. In the eighties, PICON was designed to reason in real time about process control data. It led to the development of the G2 expert system generator (developed by GENSYM [13]) which is used in many industrial applications, and in particular, by IFP (Institut Français du Pétrole) in the Alexip software [14, 15]. The expert system approach was also adopted by Sollac in the SACHEM project and by France Telecom for monitoring the Transpac network [16], before the GASPAR project had started.
This choice is justified by long industrial experience with this type of tool and its relative simplicity of implementation.

The advantages and drawbacks of expert systems for alarm processing and monitoring tasks are summarized below.

So-called real-time expert system generators are characterized by the fact that they take temporal aspects into account. Consequently there are temporal primitives in the language. This enables one to express symbolic and/or numeric temporal constraints on observed events. Another consequence is the objective of guaranteeing limited response time without making the efficiency too dependent on the way the rules are written, as this would make the rules difficult to program and read. Moreover, real-time expert system generators propose mechanisms for interruptions which allow for reaction, as well as possibilities for focusing mechanisms. Most often, there is also the possibility of compiling the rule base.

A further feature of these systems is their management of a time-stamped fact base, with mechanisms for “forgetting” or storing the facts. This type of fact base enables more natural memorization of information, considerably facilitating contextual recognition of messages. On the other hand, the very existence of this memory is a source of problems due to the continuous stream of information; management of the obsolescence of facts in the base depends heavily on the application. Very often, the mechanisms to forget or get rid of some facts are included in the expert system kernel but they are not as powerful as needed. Hence the programmer must regularly clean out the fact base to avoid that it becomes overloaded.

Looking at the formalisms adopted in expert systems, it can be noted that they are well suited to express the actions to be performed when an alarm is triggered, to tune the rules so that they can take losses of information into account, or to take all sorts of heuristics into account. Interaction between the actions performed on the system or the fact base and the condition portion of the rules, however, is often non-deterministic in theory and difficult to control in practice. Furthermore, many expert systems require complete knowledge, which the expert is unable to provide. Inference engines often rely on the (strong) hypothesis of completeness of the knowledge base. One key issue then remains the acquisition of all the expertise rules.

Finally, most expert system generators allow for easy integration of tools for processing sensor data and for simulation. They also provide user-friendly, flexible interfaces. G2, for example, offers a simulator which includes equation resolution algorithms, a graphic language enabling description of diagrams, and a module allowing for integration of neural networks.

Below, we describe a few examples of French expert monitoring systems. Sollac’s Sachem project for a high-temperature furnace process control system relies on an expert-system approach using the Kool object-oriented language. Expertise was acquired from high-temperature furnace process experts in Fos sur Mer. Alexip [17] was developed at IFP for monitoring refining and petrochemical processes. It was implemented with the G2 software and tested on the Alphabutol process, and for some parts on pilot plants. In this system, explicative models (causal graphs, influence graphs) are used in cooperation with knowledge base systems.

Another example is the software that was developed at France Telecom upon the Chronos software for monitoring the Transpac network. This system is now operational. A new approach is currently studied for this application in the context of the GASPAR project, essentially because of the difficulty to update the expertise when the monitored system changes.

In summary, the delicate aspects of this type of approach are acquisition, validation and the degree of coverage of the expertise obtained (is it consistent? does it completely cover all foreseeable situations?). Knowledge is acquired from experts, implying a risk of inaccuracy, inconsistency and incompleteness. As regards validation, the techniques proposed for knowledge bases appear well suited, but they generally focus little on verification of temporal constraints. A further point is the inherent non-generic nature of this type of model. Evolution in the process can render the full set of expertise obsolete.

On the other hand, the following benefits are widely recognized: their efficiency, due to the direct association between alarms and situations; the fact that alarm patterns generally contain only those discriminating necessary and sufficient for identifying a situation; their possible use for detection, filtering, interpretation and diagnosis; and finally, the fact that all knowledge thus used can be understood by the operators and therefore can serve directly for providing explanations or justifications.

4.4. Recognition of chronicles

While expert systems base their reasoning on rules, relegating time information to the background, recog-
The definition of chronicles is based on diagrams of evolution in which time is fundamental.

Chronicles or patterns represent a possible evolution in what is observed. A chronicle is a set of on-off events, interlinked by time constraints, and whose occurrence may be dependent on the context.

In the monitoring framework, these events could be alarms referring to the supervised system as time information would enable sequencing them and even specifying time spans between two occurrences.

For example, in the following chronicle, the alarms are partially sequenced and the temporal constraint between A2 and A5 means that it can be at least one second, but no more than three seconds, between their occurrence.

There are several methods of recognition. To name only those which take a similar approach, [18] presents the problem from the point of view of compatibility of two time-constraint networks, one of which corresponds to the model to be recognize and the other, the session, relates to the constraints between the observed events.

[19] proposes a method based on complete prediction of the possible dates for each event which has not yet occurred; all these values (called temporal windows) are reduced by propagation of the dates of observed events through the graph of time constraints of the chronicle. Recognition is incremental - each event is integrated as soon as it occurs - and it is performed over a single reading of the input stream (the system does not manage a record of observed events). This method has a high-performance algorithm, partly due to a phase of compilation of the chronicles.

The approach described in [20] is based on a finite-state automaton whose transitions are characterized by the occurrence of events. This approach is extremely efficient for recognizing sequences, but its performance is affected by the introduction of quantitative time constraints or by the use of other than sequential structures. As above, the system requires only one reading of the input stream and can later “forget” the events.

Chronicle recognition has been for example used by [21] in the AUSTRAL project in order to analyze sequence of alarms emitted by substations in a French medium voltage distribution network. It is also used in the GASPAR project in order to analyze alarms issued by the network equipment in a telecommunication network.

4.5. Discrete-event models

These are models which describe system behavior in different modes (normal behavior, degraded behaviors, behavior in the case of failure). In general, they enable simulating the system step by step, and thus predicting the values of observable variables. They can therefore be used directly to detect abnormal situations by means of comparison between predictions and observations. Because of the dynamic nature of the supervised systems, a behavior mode is described by a set of states (stable or transient) and transitions between these states.

In discrete-event models, time is rendered discrete, as are the variables. The underlying formalism is the finite-state automaton; the only dates considered are those corresponding to a change of state. Petri networks fall into this category, and have been enriched (time-lag, temporal, stochastic and fuzzy Petri nets) for improved representation of evolutions in a dynamic system. Petri networks are essentially used for simulation, and, in particular, for better handling event synchronization problems. They are not yet widely in use; we might cite the use of fuzzy Petri networks for an Esso refinery in Canada and the proposal in [22] which draws on them to build a situation graph.

Another possibility is to model system operation directly using automata. This is the approach proposed by (Sampath 94) and used by [23] for monitoring the Transpac network. In both cases, the suggestion is to build the global automaton of the system by combining elementary automata associated to system components and available in a library. This representation is appropriate to simulation and detection for interpretation and diagnosis, two methods are possible. The first consists in off-line transformation of the control into a “diagnoser”, as proposed by [24]; [23] consists in simulating this control on the basis of the most common failures so as to build pairs (set of observable events, failures) which can be used in on-line recognition of patterns ([25]). It is often necessary to represent explicitly the time constraints that are verified by the changes of state in the system. One possible formalism is that of temporal automata. The approach is used, for example, to model the Transpac network in [23, 26].

4.6. Other formalisms

A number of other possible formalisms are not described here, as we have voluntarily limited ourselves to those used in the applications with which the group
5. Our applications

In this section, we describe each application the members of the “ALARM” group have been involved into in a common framework in order to highlight the common features. Tables comparing the applications on these common features are given in the section 6.

5.1. IFP: ALEXIP project

Alexip is an architecture for supervising refining and petrochemical processes studied by IFP (Institut Français du Pétrole).

5.1.1. Supervised system

The IFP has some fifty pilot plants at the CEDI (Industrial Research and Development Centre) at Solaize. These pilot plants are small scale refining units in which various tests are carried out. The aim is to obtain very accurate results, for all the useful parameters of a process, under given operating conditions. Each pilot plant is equipped with a PC with local control software. Data are then fed back into a real-time object database to be displayed to the supervisor.

Furthermore, the IFP sells industrial units. The objective is thus to obtain products in quantity to the required specifications.

5.1.2. Aims of the supervision

One person permanently works as the supervisor over the pilot units. They detect any anomaly in any of the units and coordinate the work of the operators forming part of the crew stationed there. For this, they must monitor the changes in the different variables and must deal with any alarm reaching them. Threshold values are set on each magnitude according to the desired operating conditions. When any of these values are exceeded, an alarm is triggered which is fed back to the central station. The operator must analyze the situation and react according to the degree of seriousness of the underlying problem.

The object of the supervision system is to help operators control the processes in the optimum manner. Since control systems are installed on the units, and the supervisor has access to all the digital data available on the processes, the primary objective of an additional computer system is to arrange and filter the alarms presented to the operators so that they are not drowned in a mass of information. The aim of this alarm processing is to organize matters, not to carry out a precise diagnosis of a critical situation. This precise diagnosis requires the use of physico-chemical knowledge of the process and forms the second level of a sophisticated system of supervision. Finally, the next objective is to advise the operator on what he must do, i.e. it must help with the on-line application of the expert-defined operating procedures.

5.1.3. Inputs

On average a hundred or so variables are monitored for each pilot unit. These consist of pressures, temperatures, flow rates and possibly analyses, i.e. product composition measurements. These data are processed at the rate of one second locally and one minute centrally. At the supervisor level, an alarm is set off when a threshold is exceeded by a given variable for a certain time. Two threshold levels are defined for generating:

- safety alarms which trigger automatic safety systems;
- control alarms which are used to control the unit within the desired operating ranges.

5.1.4. Models

5.1.4.1. Causal graph for alarm filtering. For filtering the alarms occurring over time and displaying in real time the main problems to be resolved in an orderly and synthetic manner, a causal network combined with a system of priorities is used. The causal network is generated automatically off-line, from the status display of the unit. It is then examined in real time, to determine the problems at the alarm sequence source [17, 15]. The method consists in:

- choosing an entry point on the status display (a feed);
- examining each element and determining the problems which may disrupt its operation;
- examining the consequences these problems have on the elements situated upstream and downstream;
– connecting the problems with each other by carrying out pattern matching on the consequences of one problem and the conditions of occurrence of another. Delays are introduced into the links to take into account the fairly slow response times of the refining processes.

5.1.4.2. Causal graph for the diagnosis of critical subsections. Secondly, it is necessary to go deeper into the diagnostics of the sensitive parts of the process. The problem is to determine precisely which events govern the progress of the procedure. This detection is difficult due to the lengthy response times of the unit to elementary disruptions. The performance of the process is the continuous result of the combination of actions spread over time. The computer system contains the description of each event and all its consequences when it alone disrupts the operation of the unit. The consequences concern the position of the variables with respect to a given reference state and changes in them. In real time, the Dominant/Masked algorithm [28, 29] detects the dominant events for which all the long or medium term characteristic consequences are observed, and the masked events for which certain consequences are not observed due to the presence of dominant events.

5.1.4.3. Graphs of situations and operating procedures. When the diagnosis is finished, the system has the information necessary for characterizing the overall situation of the process. Each situation has an associated operating procedure [22]. A mechanism derived from Petri networks can be used to manage the chaining of situations over time. Several transitions are grouped together thanks to the concept of situation classes. The expert-defined operating procedures are described in the form of decision trees using condition or premise boxes, and action boxes. The premise boxes test the changes in certain variables, diagnostic conclusions or specific rules. Actions can be actions on operating parameters (methods of calculating their amplitude are then attached to them) or maintenance actions. These trees are used in real time to select possible action plans and arrange them in order of preference.

5.1.4.4. Causal graphs of short term interactions between variables. An explanatory module of the behaviour of the process [28, 15] uses causal graphs of short term interactions between variables. The variables are interconnected by links bearing the + sign when they are evolving in the same direction and bearing the ≠ when they are evolving in the opposite direction. Thresholds are set at which certain phenomena appear. According to the actual progress of the process, the valid causal pathways in the graph are identified to provide an explanation of the behaviour of the process.

5.1.5. Outputs

The alarm management system displays the source problems of an alarm sequence. However, certain consequent problems may be very important and demand immediate action. A system of priorities is used to identify them and display them even so. The combined use of priorities and the causal graph enables the seriousness of possible future alarms to be foreseen and, accordingly, processing priorities to be defined between the separate alarm groups. In the same way, the conclusions of the dominant/masked algorithm and the plans of actions that are generated, are displayed with their explanations according to their priorities.

5.1.6. Implementation

Alexip was developed with the aid of Gensym’s G2 software [13]. The Alexip architecture fully uses the object and graphical language offered by G2. All the algorithms were implemented in the form of G2 procedures. These manipulate the graphics containing the knowledge relating to the procedures studied. The knowledge is therefore well isolated and may be easily updated or modified by the experts. The inference engine is used for triggering the algorithms in real time on suitable data, and for focusing on critical parts of the application. Finally, “bridges”, i.e. external communication modules, provided by Gensym, are used to connect up with the supervisors and make the application independent of the latter.

5.1.7. Conclusions

Alarm processing involves all the data and calls upon the instrumentation knowledge of the process, not its chemistry. This module has been successfully tested on several pilot plants using different procedures. The proposed approach truly allows the operator’s load to be contained. It takes into account the whole of the industrial mechanism; which is useful since one finds that many malfunctions occur in the various components of an industrial site and not always in the sensitive subsections forming the subject of mathematical modeling.
5.2. Electricité de France (Network Study Branch): AUSTRAL project

5.2.1. Supervised system

The French medium voltage (MV) power distribution system is a three-phase network mainly operating at 20 kilo-volts. It is fed by the power transport and repartition networks (high voltage HV, more than 60 kilo-volts) and supplies MV/LV transformer substations and some industrial customers (both MV/LV substations and industrial customers are called loads).

At every moment, the system functions with a radial structure from HV/MV (primary) substations to the loads. Nevertheless this structure is meshable thanks to a set of remote-controlled or manually-operated circuit-breakers. This allows a reconfiguration of the system to recover a maximum of loads after a permanent fault occurs.

In a primary substation, the MV produced by HV/MV transformers feeds a busbar via an incoming feeder. The busbar supplies with MV a set of (outgoing) feeders, each of them is protected by a circuit-breaker. Fig. 2 describes a primary substation and a part of the distribution network downstream one feeder.

The objective of the system operation is to eliminate or reduce the impact that the different faults occurring on the system could have on the quality of service.

Except for some mechanical faults in the HV/MV transformers, most of the faults are electrical short circuits (more or less resistive) that may occur in the different components of a substation (transformers, incoming feeders, busbars) and on the lines of the network, for instance the appearance of an electrical arc between one phase and its support after a thunderbolt. Fault detectors (protections) are positioned in the substations as well as on the network. When a fault is detected by a protection relay, it can trigger some automatic device in order to isolate or eliminate the fault.

The main automatic devices are described below (the number of items refers to Fig. 2). As the same fault is detected by all the protections upstream, the functioning of the automatic devices is coordinated, through their specified times, so that they react in the following order:

1. some outgoing feeders are protected by a shunt that react by short-cutting the faulty phase during some hundred milliseconds in order to eliminate transient faults;
2. on some outgoing feeders (in general aerial feeders), an automatic recloser applies one or several circuit opening cycles. If the fault is not eliminated after these cycles the feeder circuit breaker definitely opens;
3. on the busbar an automatic device opens the surrounding circuit breakers (incoming and outgoing feeders, switched busbar circuit breaker) in case of an internal fault;
4. incoming feeder circuit breaker opens when a fault is detected by its associated protection;
5. when a fault occurs in an HV/MV transformer, the surrounding circuit breakers (incoming feeder, HV transformer feeder) are automatically opened.

Each of these automatic devices are fired by one or several local protections. The fault detectors and automatic devices send remote signal to the telecontrol system in the centre with respect to their behavior (fault detection, opening or closing of a circuit breaker, etc.). Automatic devices can be affected by outages, for instance, a circuit breaker may not open when asked to do so.

The French distribution system is controlled by about a hundred control centres. Each centre is responsible for several tens of substations.

The configuration is variable from one centre to another. Centres can differ because of the network structure or the equipment configuration. As an example, the centre of Lyon controls 2300km of lines and feeds 6350 loads (among them, 4000 are MV/LV substations that feed 550000 LV customers).

Each day, network operators must handle several tens of faults. Each fault generates several tens of remote signals. Around 5% of the faults are permanent and need network reconfiguration.
5.2.2. Supervision problem

The occurrence of a fault results in the functioning of several automatic devices that generate a flow of time stamped remote signals. The operator should then react in a minimum delay. He has to interpret the flow of incoming events and alarms in order to make an accurate diagnosis: what has happened, where is the fault located and which customers are de-energized? On the basis of this diagnosis, action is taken: remote switching orders are sent to isolate the fault and restore power for the maximum number of customers. A team is sent into the field for fault repair. During the early steps of this procedure, time is critical. Unfortunately, the large amount of events and alarms coming from the network during an outage may make the diagnosis task rather difficult. Moreover, reconstruction of a coherent explanation from remote events may require a fairly good knowledge of automata to link it to the actual state of network topology. AUSTRAL attempts to assist the operator throughout this procedure.

The alarm processing function of AUSTRAL (ESF) is the first function triggered after a fault occurs. The first objective of the ESF is therefore to reduce the total amount of data presented to the operator. To achieve this, sets of coherent remote signals are combined to form a single synthetic data entity. The second objective of ESF is to provide a fuller analysis of incoming events, in terms of outage diagnosis. Around 20 types of diagnosis have been identified. They correspond to a synthesis of correct behavior or misbehavior of automata and protection devices.

According to the diagnosis produced by ESF, other AUSTRAL functions may be launched. FLF (Fault Location Function) will be triggered when a fault on an outgoing feeder occurs; its objective is to locate the fault on the network by consulting fault detectors. NRF (Network Restoration Function) will be activated for every permanent fault; this function proposes to the operator a reconfiguration plan (a list of open/close circuit-breaker commands) in order to restore power for the maximum number of customers.

AUSTRAL is connected in real-time to the telecontrol system. A full description of the AUSTRAL package can be found in [31].

5.2.3. Inputs

When a fault occurs on the network, captors trigger some automatic devices and inform the operator (via remote signals). The principle of captors is to measure currents or instantaneous powers and to compare these measures between themselves or to thresholds.

The detection of a fault results in the functioning of automatic devices (shunt or recloser cycles, opening or closing of circuit-breakers) that generates remote signals according to their behavior.

The supervision system inputs are the set of remote signals send by the different network equipments. For a network like the one of Lyon, around 8000 different remote signals are possible; these signals are gathered into around 20 types.

The informations coming from protections and automatic devices in the substations are considered to be reliable and are used by ESF to perform fault diagnosis.

The informations coming from fault detectors on the lines are uncertain (unreliability of equipments, possible communication problems). They are used by the FLF to locate faults.

5.2.4. Models

Two kinds of models are used:

- a set of generic chronicles describes the classes of different distribution system behaviors after a fault has occurred. Chronicles are used by the ESF for fault diagnosis. Each chronicle is associated a synthetic message that will be displayed to the operator when the chronicle is recognized (see section 4.4). The complete knowledge base contains around 100 chronicles. Each chronicle corresponds in average to some tens of events.

- a dynamic database (RDTS for Real Time Data Structure) reflects the state of the whole network at every moment. The RDTS is updated by the telecontrol system and is used by all AUSTRAL functions.

The three main functions of AUSTRAL use these models in different ways.

- The diagnosis function (ESF) performs chronicle recognition and generates a synthetic message when a chronicle is recognized.

- When a permanent fault occurs, the fault location function (FLF) consults fault detectors on the faulty feeder and locate the fault downstream the last detector that has seen the fault and upstream the detectors that have not seen it. AUSTRAL makes hypothesis in order to minimize the number of detector outages when the informations coming from fault detectors are incoherent.

- The network reconfiguration function (NRF) uses the dynamic database to generate a set of reconfiguration plans. AUSTRAL weights the generated
plans thanks to an aggregated quality criterion that takes into account the customers de-energized, the easiness to execute the plan and its robustness.

The network dynamic database used by AUSTRAL is derived from the telecontrol system’s one. The connection of AUSTRAL to the telecontrol system via API (Application User Interface) was one of the main difficulties of the programming part of the project.

The chronic knowledge base for the ESF was collected and tuned by a network expert during around one year, in collaboration with the final users.

Because of the number and complexity of chronicles, the problem of validation and evolution of the knowledge base is a bottleneck. We are developing a tool, called GEMO, to automatically generate and validate the chronic knowledge base from a deep model of the network equipments using communicating automata [32, 33].

5.2.5. Outputs
There are two kinds of outputs for AUSTRAL.
- Firstly, a display window (Synthetic Summary) informing the operator about currently happens on the network. The messages displayed in this window are the ones generated during chronicle recognition by the ESF.
- Secondly, when a permanent fault is diagnosed, a list of reconfiguration plans - sorted according to the quality multi-criterion - are proposed to the operator.

5.2.6. Implementation
The different AUSTRAL functions are implemented as independent process and are executed in parallel as it would not be realistic to stop the event analysis and diagnosis function because a reconfiguration plan must be computed by the NRF.
AUSTRAL functions can query the RTDS (for example about the network topology at a given moment), send queries to the telecontrol system (for example in to interrogate the fault detectors) or receive events from the telecontrol system (remote signals).
AUSTRAL works on Sun and IBM architectures.
AUSTRAL functions are written in ILOG Talk (Object-oriented Lisp compilable in C). The RTDS as well as the interface between AUSTRAL and the telecontrol system have been implemented in C++.

5.2.7. Validation, maintenance, evolution
At present, the tool is being tested in the centre of Lyon. Three French centres will be equipped with AUSTRAL in 1998. The potential customers are all the distribution centres in France, as well as some centres abroad.

5.3. Electricité de France (Monitoring, Diagnosis and Maintenance Branch): DIAPO project
DIAPO [34] stands for Diagnosis Support of Reactor Coolant Pump Sets of Pressurized Water Nuclear Power Plants.

5.3.1. Monitored System.
The Reactor Coolant Pump Set (RCPS) is a major component of a pressurised water reactor: it ensures circulation of the coolant fluid between the reactor core and the steam generator (figure 3). Its good functioning is decisive for plant availability (stopping the RCPS results in plant shutdown) and security (primary fluid also serves to cool the reactor core).

An RCPS is about 10 meters high and 80 tons heavy with a flow rate of 7 m$^3$/s. Rotation speed is of 1500 rpm. Three functioning phases are considered: startup, nominal operating conditions and rundown.
Experts generally distinguish the following main functional assemblies (figure 3): the pump subsection, where an impeller drives primary circuit water; the thermal barrier and the seal system, preventing primary coolant water from rising; the shaft line, consisting of three coupled shafts whose stability is ensured by bearings; the motor, which ensures the shaft line rotation and a flywheel which moderates slow-down time in case of electrical power loss.

About 150 RCPS operate in the French nuclear power plants. Their expected lifetime is over 30 years. Three different RCPS technologies are distinguished, corresponding to the three steps of conception of the nuclear power plants. Some differences can also exist between RCPS of the same type, due to local technological modifications. However, these differences are worthless, with respect to the diagnostic problem.

The RCPS is a dynamic system; still, its characteristic variables (flows, vibrations, temperature...) are almost stationary during nominal operating conditions.
RCPS failures have various dynamics: from brief phenomena (e.g. a part loss) to very slow ones (e.g. progressive loosening of a bearing fixation). RCPS behaviour is altered during its life-time, because of its components wear and aging.
5.3.2. Surveillance and diagnostic aims

Final users of DIAPO are engineers of the RCPS maintenance staff, who have to decide whether the RCPS must be stopped or not in case where an abnormal behaviour is detected. This responsibility is quite important as the RCPS run-down entails the plan unavailability. In addition diagnosis and surveillance of the RCPS are expected to support optimization of its maintenance.

The expected frequency of use of DIAPO is low, RCPS being reliable machines (between 1 and 10 times a year, depending of the plant). It is worth noting that the RCPS is not monitored in order to optimize its functioning.

5.3.3. Diagnostic Inputs

The RCPS is continuously monitored by an automated system using some 30 sensors to characterise the condition of the machine and generate alarms.

Monitoring data are stored in a data-base and can be retrieved for real-time or delayed consultation. High-level parameters are computed from “basic” sensor data (e.g. for measured vibrations, amplitude and phase of each harmonic).

Altogether, about 200 observable parameters may be used to describe RCPS states, including: monitoring system data, such as thermo-hydraulic (bearing temperatures, seal flow...) and vibratory measurements (shaft and housing vibrations...); plant state parameters, (primary circuit temperature...); locations where significant observations may be performed (under the engine, in bearing lubricant...) and maintenance and control operation logs.

It is worth noting that the diagnostic system is not automatically triggered by the monitoring system.

5.3.4. Models

5.3.4.1. Types

DIAPO diagnostic method strongly relies on advanced research works on abductive and temporal reasoning, especially [3, 4, 35]. DIAPO uses four fault models described figure 4.

- A Prototypical Fault Model which produces hypotheses on the abnormal situation encountered.
- A Localization Fault Model which produces hypotheses on the location of faulty components.
- A Causal Fault Model which produces hypotheses on the primary causes of the RCPS misbehavior.
- An Associative Causal Model which is a simplification of the deep causal model, consisting of direct cause-to-effect relations between a hundred RCPS faults and their manifestations.

In these models, symptoms play a particular part, as their truth value can be established through RCPS observation. They consist in predicate calculus formulae with existentially quantified variables. For instance, the symptom There exist a variation of type step with amplitude greater than 25 microns of the
monitoring parameter DA is represented by the formula: \(\forall(x)[\text{Type}(x, \text{Step}) \land \text{Parameter}(x, DA) \land \text{amplitude}(x) > 25\mu]\). In each diagnostic problem instance, variables will be assigned by events describing the RCPS behaviour, in order to compute the symptoms satisfied by the actual RCPS observation.

These events are the individuals which constitute the RCPS observation for a diagnostic problem instance. They are defined by:

- a parameter \(P\) representing the monitoring measurement involved (e.g. vibration measurement; seal flow...)
- a type \(T\) representing the property verified by the parameter (e.g. step, abnormally high...)
- a fuzzy temporal extent defined by a beginning interval \([b_1, b_2]\) and an end interval \([e_1, e_2]\) where \(b_1, b_2, e_1, e_2\) are time points verifying \(b_2 \geq b_1, e_2 \geq e_1, b_1 \leq e_1, e_2 \leq b_2\) and on which the property represented by \(T\) holds.

For instance, the observation A 2 second long step on the vibration measurement was seen on April 2\(^{nd}\) at 12:00:00 will be represented by the event:

- parameter: vibration measurement
- type: step
- beginning interval: [April 2\(^{nd}\) 12:00:00, April 2\(^{nd}\) 12:00:00]
- end interval: [April 2\(^{nd}\) 12:00:02, April 2\(^{nd}\) 12:00:02]

This representation allows to deal with observations whose beginning and end are not known with precision. For instance, the observation The seal flow is abnormally high (noticed on April 1\(^{st}\) at 12:00) can be represented by the event:

- parameter: seal flow
- type: high level
- beginning interval: \([\neg \infty, 1\text{April} 1^{st}\ 12:00]\]

Delays attached to relations in the fault models represent general knowledge on RCPS fault dynamics. Informally, the delay \(\Delta\) in the cause-to-effect relation \(C[\Delta] \rightarrow E\) describes the relations between \(C\) and \(E\) temporal extents. When \(E\) is a symptom, such a delay may be specified for each existentially quantified variable of \(E\). Practically, it consists in constraints, possibly approximate, between respective beginnings and ends of \(C\) and \(E\). For example, let \(C = ([b_1, b_2][e_1, e_2])\) be the temporal extent of \(C\); let \(\Delta\) be “\(E\) begins between 1 and 6 months after the beginning of \(C\); \(E\) and \(C\) ends are simultaneous”. The temporal extent for \(E\) is then: \(([b_1 + (1\text{month}), b_2 + (6\text{months})][e_1, e_2])\).

In addition to fault models, DIAPDO also uses an Observation Model which consists of relations describing possible dependencies between events observed. These relations are based upon physics, mathematics, and signal processing theory laws, applying to the functioning on the surveillance of RCPS. Here is an example: a such a law:

Any high level state observed on some continuous monitoring parameter is necessarily preceded by an increase.

These relations can be used to automatically complete the observation set during problem solving. Some of them can also serve to complete the fault models in order to prevent from explaining independently correlated events.

5.3.4.2. Size. Fault models currently used by Diapo contain about 1500 relations.

5.3.4.3. Use. An abduction procedure integrating temporal information propagation is applied to each fault models. It computes formulae that explains the ob-

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Relations</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal states of the RCPs, events, symptoms, maintenance and control operations, logical and temporal operators</td>
<td>(C[\Delta] \rightarrow E)</td>
<td>(C) causes (E) after (\Delta)</td>
</tr>
<tr>
<td>(C_1 \land \ldots \land C_m[\Delta_1 \ldots \Delta_m; \Delta] \rightarrow E)</td>
<td>(C_1\ldots\land C_m) causes (E) with delays (\Delta_1\ldots\Delta_m; \Delta)</td>
<td></td>
</tr>
<tr>
<td>(C \land A[\Delta] \rightarrow E)</td>
<td>(C) causes (E) after (\Delta), under contextual condition (A)</td>
<td></td>
</tr>
<tr>
<td>(C \land \alpha[\Delta] \rightarrow E)</td>
<td>(C) causes (E) after (\Delta), under non-specified condition represented by (\alpha)</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4. Relations for Diapo Fault Models
served events and are consistent with the information that event occurrences are negated. Such an explanation takes the following form:

\[ EXP(S_i, \{ e_1 \ldots e_n \}) \equiv \]

\[ (C_{1i}, C_{1i}) \lor \ldots\]

\[ Sync(\Delta^{k,n}(C_{ik}), \Delta^{k,(i)}(C_{ip}) \land \ldots) \]

\[ (C_{il}, C_{il}) \land (\alpha_{ij}, \alpha_{ij}) \land \ldots \lor \ldots) \]

\[ \land \neg(C_{k_i}, C_{k_i}) \land \ldots \neg(C_{k_i}, C_{k_i}) \]

Where:
- \( S_i \) are symptoms, \( \{ e_1 \ldots e_n \} \) their satisfying assignments
- \( C_{ij} \) are initial-causes; \( C_{ij} \) are their temporal extent.
- The \( Sync \) predicate represents the synchronization condition attached to conjunctive terms. \( \Delta^{k,n}(C_{ik}) \) represents the application to the temporal extent of \( C_{ik} \) of the series of delays held by the relations between \( C_{ik} \) and the conjunction.
- \( \alpha_{ij} \) are abstract conditions, \( \alpha_{ij} \) their temporal extent.
- \( C_{ij} \) are the refuted terms, and \( C_{ij} \) their temporal extent on which they have been refuted.

Diagnosis is each model is obtained by computing the best conjunctions of such formulae according to heuristic criteria (maximum of events explained, best circumscription of explained events...)

Final diagnosis is the conjunction of the diagnoses obtained in each fault model. Its worth noting that an hypothesis assumed in one fault model can be refuted in another fault model, if the latter is more complete that the former.

5.3.4.4. Acquisition. Knowledge acquisition required five experts and two knowledge engineers during about one year. It was performed without any specific knowledge acquisition tool.

5.3.4.5. Limits. The main limit of this diagnostic method is its incompleteness: DIAPO can produce diagnoses only for cases embodied in its model. This limit is inherent to any fault model based diagnostic method. This limitation can be severe, when facing multiple faults. Indeed, to limit complexity and combinatorial character of the models, failures are considered independent. That is, two independent causes are assumed to produce the conjunction of their respective effects, except an explicit description of their combination is present in the model. But this hypothesis is not always sound. Two independent failures occurring simultaneously can interact, resulting in some unexpected effect not described in the model. In this case, diagnosis will fail.

5.3.5. Diagnostic Outputs

Outputs of DIAPO consists of:
- the diagnosis obtained.
- the causal paths between initial-causes and the explained events
- the justifications of the refutations

Complementary information attached to term specified as faults can be displayed: gravity level, maintenance advice. This outputs can be browsed through a graphical interface, and saved in a printable format in order to produce diagnostic reports. It is worth noting that the diagnostic system has no feed-back on the RCPCS process.

5.3.6. Implementation

The diagnostic process is decomposed into several tasks (problem solving in one particular model, complementary observation acquisition...) and implemented in a blackboard architecture.

The whole system is implemented in Lisp, in the Illog’s development environment SmeciTm. A full-scale prototype a been developed in co-operation between EDF and the RCPS manufacturer, Jeumont Industrie, for a global cost of 10 Million francs, including the acquisition and the validation of the knowledge bases, the development and the test of the prototype.

5.3.7. Perspectives

From a technical point of view, one important perspective is to provide DIAPO with consistency-based diagnostic method, using a model of the normal behaviour of the RCPCS. The use of heuristic knowledge in order to focus abductive reasoning is also studied; as well the introduction of probabilistic information within the fault models.

The industrialization of a simplified product derived from the prototype is currently under development, in cooperation with Jeumont-Industrie, the RCPS manufacturer. This system will be installed in 20 EDF sites and proposed by Jeumont-Industrie to its foreign
clients. The direct expected benefits for EDF are evaluated to 2 million francs per year, through avoiding of power plant shutdowns enabled by DIAPO diagnoses of RCPS faults at incipient stages.

5.3.8. Validation, maintenance, evolution

Validation issues concern the Fault Models and the Observation Model. Since formal methods lack to validate this knowledge, a hundred tests have been performed in order to evaluate the prototype. About 10% of these tests were real incident reconstructions, the other part consisting in theoretical situations built by the experts and the developers. 80% of diagnoses were considered satisfactory by the experts. Failures were due to incompleteness of the fault model.

The tests have shown a good evolutivity and maintainability of the system, thanks to the unified scheme of resolution employed: the structure of the fault models allow to modify them easily.

5.4. France Telecom : GASPAR project

5.4.1. Telecommunications Networks

French packet switching data transmission network includes two independent entities, with respect to operation, which are the transport network, ensuring connection setup and data transfer (voice, text, images), and the management network, which transmits orders to network equipment, and receives alarms which are then routed to the supervisor.

The architecture described in figure 5 is organized in tree-structure between switches (networks nodes), the technical centres and the supervisor. About fifty techniques centres, which group over three hundred switches, are controlled by the supervisor.

Each equipment in the network can happen to fall out of order and each equipment in the network can be put back to nominal functioning conditions through reinitialization or switch over to standby equipment, which means the network is “protected”. This kind of network operates non-stop, hence supervision must also be ensured round the clock.

As faults as well as coming back occur, alarms are sent by equipments to the supervisor through the network. Masking phenomena may occur with this type of architecture. If a component of the tree-structure has failed, all the trees of the sub-tree will be unreachable and in a unknown state, hence they will not be able to notice the supervisor (even everything work fine at their level). Such loss of alarms must be taken into account when making a diagnosis.

5.4.2. Objectives of Supervision

Continuous supervision of the network makes it possible to be informed at any time about the state of the network, e.g. which device is backed up or out of order. It also allows to know the origin of the failures, without having to check the various equipment or to go over all the alarms. The important thing is to screen the alarms issued by the network (about 150 000 per day) and interpret them. In this way, only events of major importance are notified to the operator.

Supervision is particularly useful for short-term network maintenance. As a matter of fact, operators have to evaluate the extent of the problem and initiate corrective actions or failed components. The equipment includes automatic control facilities which take the functional aspect of the network into account.

5.4.3. System inputs

System inputs mainly consist of alarms issued by the network equipment and routed over the tree-structured supervision network. Data may also include responses to queries about the state of a particular equipment, put by an operator or by the supervision system.

These alarms are symbolic since no quantitative information is given regarding the traffic density or switch load. They include information about:

- the component state change (e.g from its normal state to out of order and vice-versa)
- connection breakdown between two pieces of equipment (e.g. between a technical centre and a switch),
- reinitialization of a component (e.g a switch).

The alarms are reliable but two phenomena may induce losses of alarms:
The ALARM research group / Monitoring and alarm interpretation in industrial environments

- saturation of buffers: alarms are buffered in each switch, but switching takes priority over their processing; when buffers are full, further alarms will be lost.
- masking of alarms: in the event of component failure, the alarms sent by the subtree to this component are lost, hence they will not be received by the supervisor (figure 5).

At least, one hundred different alarms can be observed in the various parts of the network. The number of alarms has significantly increased in the last few years: switches of the last generation can produce as many as 150,000 alarms per day. Another difficulty is that of propagation of alarms. When a technical centre breaks, it causes the booting of all its switches (hierarchical dependency). Thus, a technical centre breakdown will be responsible of the emitting of many alarms (as many as switches).

It should also be noted that buffers placed along the alarm feedback paths induce variable delays between transmission and reception of messages. Hence the correlation between alarms from different paths may be more difficult.

5.4.4. Implementation

The present knowledge was devised by an expert, it is organized in about two hundred deduction rules. It was so costly that it was decided that network evolutions would not be taken into account.

System outputs are alarms, failures contexts (reinitialization, connection breakdown...) and information of the state of the network (the diagram is updated using change-of-state alarms).

This system has been in operation for several years. However, expert knowledge procedures have been stopped and the operators did not propose or wait for further evolutions because of insufficient real-time performance and also because, nowadays, network expansion is the highest priority task for the operators, hence knowledge acquisition is delayed. Attention is still focused on network supervision but aids will have to be developed for expert knowledge acquisition and evolution.

Considering these problems, it appeared necessary to go deeper into model-based reasoning techniques and also training and data mining techniques. These latter are still at a prospective stage, however.

5.4.5. The Gaspar project: knowledge acquisition by simulating models

The GASPAR project aims to study the application of artificial intelligence to telecommunications network supervision. Particular emphasis is put on model-based diagnosis and to a greater extent on qualitative reasoning.

One of the advantages of model-based diagnosis is generics, but this method may be costly (in execution time). For this reason, the architecture of the GASPAR system is organized in two parts (see figure 6): network modeling and failure simulation are performed off-line [36, 23]; it enables characteristic sequences (called scenarios or chronicles) to be obtained. Detection of the sequences is then performed on-line, using tools such as IxTeT [19, 37].

5.4.5.1. Modeling. Modeling takes advantage of the formalism of communicating temporal automata adapted to discrete event systems; this method is suited to the construction of a model combining simple elements [38]. Two levels of abstraction can be made out in this construction:

- The connectivity network level, which describes the transmission of messages, giving greater importance to the transmitting or non-transmitting state in each node. The model is not much different from that presented in figure 5.
- The description of each component behaviour in automaton form. The transitions between different states correspond to reception or transmission of messages. Some of these messages correspond to alarms sent to the supervisor.

The French acronym GASPAR stands for “Gestion d’Alarmes par Simulation de PAnnes sur Réseau de télécumunication”

2collaboration between France Telecom/CNET and LIPN/CNRS and IRISA/CNRS
5.4.5.2. Simulation. Simulation is made taking failures as a basis, using the behavioral model of a network with “failure events”. The time division aspect of component functioning is also taken into account as well as the uncertainty regarding the time of occurrence of some events. Simulation thus reproduces the propagation of alarms to the supervisor. Masking phenomena are also simulated (for instance when a technical centre is in non-transmitting state during the routing of a message issued by a switch). The sequences of alarms received by the supervisor for a given series of failures can then be worked out.

5.4.5.3. Generics. For a lot of network configurations, the method applied is generic, since the global model is obtained from a number of elementary models. If a modification or addition of a component is necessary, the component library is updated and the global model is easily reconstructed. In this way, maintenance procedures are facilitated.

In addition, the time aspect is taken into account for component functioning which enables transmission durations to be represented.

5.5. CEA Marcoule: DIAPASON project

DIAPASON is a system developed by the “Commissariat à l’Énergie Atomique”.

5.5.1. Supervised System

Spent fuel is reprocessed to recover the uranium and plutonium still present in the fuel rods irradiated in nuclear reactors, and to isolate the remaining fission products. The pulsed column facility considered here includes extraction columns and fission product scrubbing columns, and is designed to separate the uranium and plutonium from the fission products (FP) by selective extraction. A pulsed column is a liquid-liquid extraction device. The spent fuel (comprising uranium, plutonium and FP) is dissolved in nitric acid, and the extraction column selectively transfers the uranium and plutonium to an organic phase consisting essentially of tributylphosphate (TBP); most of the fission products thus remain in the aqueous phase. The extraction step requires that the aqueous and organic phases be thoroughly mixed to maximize the contact surface area between the two solvents and thereby optimize the chemical exchange phenomena. The two phases tend to separate by gravity, as nitric acid has a higher density than TBP.

In order to ensure countercurrent flow, the light (TBP) phase is injected at the base of the column and the heavy (acid) phase at the top. The organic phase is the continuous phase; it initially fills the entire volume of the extraction column, and remains the predominant phase after injection of the aqueous phase: acid droplets are dispersed in the TBP. The resulting mixture is subjected to periodic pressure pulses to form an emulsion in order to retard the descent of the heavy phase and to mix it with the light phase. The interphase the surface physically separating the two phases is located in the settler at the bottom of the column, and is regulated by drawing off a the aqueous phase at a suitable rate. In the application considered here, all liquid transfers are ensured by airlifts.

DIAPASON was developed and tested by means of a process simulator based on numeric codes used by the DCC/DRDD/SEMP: numeric model simulations constitute the “actual” or “measured” process behavior. A total of about seventy failures were modeled. A control interface is provided for process control and for implementing failure modes.

5.5.2. Surveillance Objectives

Nuclear processes are prime candidates for supervisory aids:

- The process media are highly radioactive and it is thus difficult to install sensors that are both sophisticated and durable. The measurement equipment is thus implemented using rudimentary techniques with minimum maintenance requirements. Chemical measurement sensors are particularly difficult to develop; as a result, much of the data concerning chemical phenomena are obtained off-line by sampling.
- The physical and chemical phenomena involved are relatively complex, and process control requires highly experienced operators; moreover, many tests are performed on prototypes to assess the effects of various parameters on process behavior.
- Process control is also highly sensitive since the process units are interconnected to allow continuous product recycling, and the equipment items in each unit are closely coupled.

DIAPASON provides the operator with a synthetic representation with modeling based on qualitative physics and automatic control [39]. It is designed to predict

process evolution to allow anticipative control. In incident situations it detection, localization and diagnostics of malfunctions for preventive maintenance or for process control in degraded mode.

5.5.3. Inputs

All inputs are digital values, with no descriptors of any other type. The digital simulator is seen as a black box whose outputs constitute the sensors used by DIAPASON for its surveillance functions. Noise may be added to each digital output line from the simulator to simulate measurement noise.

The objective is to provide the operator with information strictly related to diagnostics; only pertinent changes in variables (i.e. changes liable to affect overall process operation) are therefore taken into account. Segmentation is used to convert the sampled process input signal (a computerized representation) into a series of significant variations (a more explicit representation for the observer). Changes in measured process variables are described as a series of significant events represented by a segmented affine time function. A change in slope corresponds to a significant change in the behavior of a variable. In order to allow for instantaneous phenomena (at the supervision time scale) these functions may be discontinuous. An event is thus characterized by a triplet occurrence date \( t_0 \), slope variation at \( t_0 \), amplitude variation at \( t_0 \). The evolution is thus perceived as a succession of chronologically ordered events.

5.5.4. Models

The model is an influence graph in which the vertices are the process variables and the arcs are the causal relations among the variables (PROTEE module)[40]. The evolution of a variable is described numerically and an arc (qualitative transfer function) characterizes the influence of one variable on another using basic automatic control concepts (dynamic allowance for influences). The overall process evolution is characterized by the state of the pertinent variables: a significant variation of an input variable corresponds to the detection of a graph source event; the propagation of the event in the graph indicates the process response to the input.

Each arc on the graph defines the temporal causality relation between two variables using conventional notions to provide a dynamic description of the influences (gain, pure delay, response time, etc.). By analogy with classic automatic control systems, the function supported by the arc was designated the Qualitative Transfer Function (QTF).

The QTF response to an input signal is approximated by a segmented affine function from the response of a conventional transfer function to the same input, i.e. it is an “evolution”. The simulation consists in propagating significant changes affecting the graph sources (i.e. the process inputs) from one variable to another using QTFs. The event-driven nature of the simulator is due to the fact that these changes do not occur at regular intervals.

The graph for the pulsed column facility comprises about 55 variables and 70 arcs. The event simulator providing the process “reference” behavior is obviously less precise than the digital simulator used to obtain the “measured” behavior, but the development cost is of another magnitude even if the construction of the causal graph is not a simple task and requires the cooperation of an expert. The models declarative nature simplifies revisions.

Detection (MINOS module [41]) involves local processing in which each deviation between process measurements and the data simulated by PROTEE is analyzed over a specified time interval. The simulations better represent process behavior than exact numeric variable values; a simple comparison with a single-threshold error is thus inadequate (inappropriate thresholds, noise sensitivity, etc.). We therefore introduced the notion of qualitative equivalence between two evolutions to provide the module with greater insensitivity to noise (modeling approximations, measurement noise, etc.). Various comparison criteria are considered: curve shapes, deviation, distance between curves.

In view of the multiple sources of inaccuracy, we have attempted to model vague and imprecise phenomena by implementing the theory of fuzzy sets [42]. Each criterion requires a symbolic interpretation leading to a decision.

The detection phase is followed by a localization phase (MINOS module), an overall analysis of all the simulation errors in the graph. It constitutes a solution to the alarm cascades by proposing causal chains linking the deviations affecting all the variables in the graph. Dynamic management of the fault coherence yields the propagation subgraph of a trend and identifies the possible source variable(s).

A source variable is defined as the first variable (in time) for which the influence of a perturbation is observable. A detection variable is a variable for which the deviation between the actual evolution and the expected evolution exceeds the permissible threshold. The possibility of tracing back from a detection variable to a source variable compensates for the often arbitrary nature of the threshold technique.
When faults are detected, the graph is used for dynamic investigation of the errors in analysis time windows. Tracing the qualitative graph from effect to cause produces a propagation subgraph for the error and its possible sources.

The coherence test consists in substituting real events for predicted events in the analysis time window for each variable preceding the detection variable. Propagating these new events yields a new simulated evolution for the detection variable, and subsequently a new model/process error on this variable, which is compared with the initial deviation: a qualitative interpretation (i.e. deviation eliminated, unchanged or reversed) of the error variation between the two simulations is based on orders of magnitude, and provides information on the causal links between the upstream deviations and the deviation of the detection variable.

This coherence test can be interpreted as an alarm filter: before reporting the occurrence of a new defect, the causality analysis attempts to relate the new alarm to the previous suspect subgraph. We have also shown that in the case of detection this defect filtering capability could be modeled as a decision-making process applied to the aggregate of criteria described by fuzzy sets.

The failure diagnostics (SPHYNX module) implement structure knowledge of the process to identify the defective component. Fault diagnostics performed by the behavioral model are not generally sufficient to identify the physical component failures associated with these faults, as the processing would require structural and functional knowledge not included in the model. However, a knowledge-base system is poorly suited to time management; adding temporal logic would complicate the rules and raise problems in validating the knowledge base. We therefore decided to have the failure diagnostic expert system manipulate the expertise, and assign time management considerations to the MINOS module, which activates the expert system only after a defect is observed, by supplying an image of the process formed by actual and predicted source variable values at the instant t when the default is detected, as well as the values of the other variables at coherent prior times (allowing for the delays inherent in the causality relations). This approach simplified the diagnostic system task, thereby enhancing performance.

5.5.5. Outputs

The event-driven simulator yields the behavior of all the causal graph variables in the form of segmented evolutions. A predictive horizon may be specified by the user. The actual and predicted evolutions of variables that can be selected on the graph can be monitored on child windows. Three types of alarms are indicated for each graph variable:

- a predicted alarm: the variable at the operator-selected prediction time horizon exceeds an alarm threshold;
- a pre-alarm condition: the variable is currently within the permissible range, but its current value does not correspond to the simulated value (i.e. the detection module reports an alarm condition);
- an alarm: the variable already exceeds an alarm threshold in the conventional sense of the term.

The MINOS module supplies the following outputs:

- the detection system provides a set of defective variables;
- the localization system organizes this set into one or more defect subgraphs whose evolution can be monitored graphically as the propagation of disturbances in the facility: the propagation can be followed by the color of the affected arcs and variables over time: green (normal) or red (defect).

SPHYNX assigns failures to the equipment items in the monitored facility. The output is thus a set of failure hypotheses whose degree of plausibility can be followed.

Finally, an interface child window is reserved for the explanation. In order to know the explanation for the behavior of a variable, the contributions of the upstream variables are presented (using the superimposition theorem) opposite the history of the specified variable. The following information is thus available for any operator-selected variable: the designated upstream variables acting on the result; the direction of the action of each upstream variable and the comparative amplitude of each contribution. This explanation also provides a solid basis for developing new plans of action to modify the state of the variable.

5.5.6. Implementation

The three system modules were developed in Ada on a Sun workstation, with the exception of the rule compiler written in STARLET, a predicative language implementing affixive grammar. The three modules have been integrated into a demonstrator running online with a conventional numeric simulator of a nuclear fuel solvent extraction and scrubbing facility.

The prototype version is the outgrowth of three Ph.D. thesis at the LIA with the collaboration of the INSA.
at Lyon (Professor L. FRECON) and the Laboratoire d’Automatique de Grenoble (Professor S. GENTIL).
The interface (MIMIX) and interprocess communications required 1.5 man-year of work. The following knowledge is necessary to develop a new application with the PROTEE, MINOS and SPHYNX modules:

- constitution of the causal graph of the facility for the PROTEE and MINOS modules;
- interpretation of the AFME for SPHYNX.

Nevertheless, the causal graph representation allows the use of a single, simple model for simulation, explanation and localization purposes; the declarative nature of the model clearly discriminates between knowledge and reasoning.

The separation between defect diagnostics (localization) and failure diagnostics makes it possible to initiate only local studies with the expert system, while reserving time management for the more appropriate tools of the MINOS module.

5.6. Sollac : SACHEM project

5.6.1. The supervision system

The Sachem project (Computer aided blast furnace operation) is being set up at Sollac to help operators to drive the Blast Furnace (BF) at Fos sur mer and Dunkirk (France). Sollac is a subsidiary of USINOR, the first steel maker company in Europe and the third in the world. First of all, coal and coke are introduced in the BF in order to make cast iron which gives afterwards steel. The materials are introduced at the top and the middle of the furnace and the cast iron is picked up at its bottom at each casting. We can find the three phases, the solid, gaseous and liquid ones together in the furnace. Therefore the phenomena are complex with regard to thermic, chemical and mechanical energies. Their identification is rather complex too. The IRSID (the research center of USINOR Group) has already developed a few partial models, for instance one dealing with the balance of energy. But there is no global model especially no dynamic model.

There are many different BFs but we can consider that they all have the same process. Man has been able to produce cast iron for 3,000 years. A modern BF is therefore the fruit of much experience, patience and technology. A BF works continuously without stopping except for programmed stops every four months and emergencies which are unusual. The BF process is slow: when you add some coal or coke it has an effect on the cast iron temperature between four and eight hours later.

5.6.2. Purpose of the supervision

The Sachem project has been set up to help supervision of the BF particularly to improve its regularity and the homogeneity of the cast iron. This allows a longer duration of life of the BF, less operation in the steel making plant and therefore a lower cost price of the final product (the coil). Operators cannot easily reach this aim of regularity because of the delays between an action and its effect, the effect of their decision and action is usually seen by the following team. The early detection of malfunctions allows the operators to take corrective measures with anticipation. This is a key factor for a good control of the process. Another aim of Sachem consists in diminishing of the consequences of the turn-over in the team of operators for instance when they go into retirement.

Sachem is to survey the real-time process, point out the phenomena, look for their causes and recommend appropriate actions. Sachem is in operation in the control room. Of course, the operators already work thanks to data given by one thousand captors, but data are so numerous that they cannot be used at their best. There are about ten actuators to drive the BF. One of the most important problem is the thermic regulation which deals with the cast iron temperature. Moreover, a specialized function of Sachem helps the process analysis team.

5.6.3. Input

5.6.3.1. Gross data. Sachem analyses data provided by captors situated all around the BF. There are roughly 1,000 captors which give continuous signals with a frequency of a data every minute. The data are acquired by a specialized module - the acquisition module - and are tested for validity. These data gather: temperatures from the wall of the furnace (staves, heat resistant walls etc.), pressures on the wall in order to calculate the per-
meability of the BF, gas analyses: temperature, chemical analyses...flows: flows of gas at the nozzle, flows of gas out of the BF, cast iron flows...

These data are afterwards sent to an elaboration module which produces 3,300 variables. The mathematical rules used within this module are for example the average, the smoothing. Sachem uses also the partial mathematical models of the BFs.

5.6.3.2. The Signal Phenomena. They characterize the previous measures and describe: the level of a signal, its stability, the variations (e.g. of the slope...), the long term tendency, the undulation of the signal...

These signal phenomena correspond to the signal characteristics that have been selected as significant by the experts. The ENQUET environment - based on the X-ANALYST software and commercialized by the AIS company, a young company coming from Matra Marconi Space - is used to tune and parameterize the signal analysis algorithms. The techniques consist in applying a treatment on a data range and in visualizing the intermediary and last results.

5.6.3.3. The BF Phenomena (BFP). They are obtained from the signal phenomena thanks to the use of coded expertise linked to a knowledge base. We have counted about 150 phenomena which induce about 450 possible different messages (alarms or warnings) for the operator.

Let us give examples of these alarms: “the increase of the gas distribution on the wall”, “a low cast iron temperature”, “a high level of slag index”...Of course, the detection of some phenomena depends on the functioning of the BF. The thresholds used by the signal analysis module are calculated from the information describing the context of functioning.

Here is an example of an expertise rule: If there is a signal phenomena of decrease of wind in a nozzle and within two hours a signal phenomena of increase of wind in the nozzle superior to 80% of the previous decrease then a BFP of punctual passage of bloc is detected for this nozzle. Consistency and validity control are effected on the data and the phenomena. If some data miss or are not valid, some other substitution data can be used. These data are therefore used by the knowledge base as long as the invalid data are not specified as operational again.

5.6.4. Models

The identification of the BFP from signal phenomena is due to a knowledge base embedding the domains expertise. It is implemented in Kool 96 - a hybrid powerful generator - and structured within 20 rubrics of expertise.

The following ones have been identified: local conditions of nozzles, quality of the cast iron, global permeability, thermic balance etc. In each rubric can be found mainly the following points:

- the issue of the rubric,
- the typical phenomena of the rubric and the detection rules,
- the invalidation rules and the substitute detection rules in case of invalidation.

Some detection rules of BFP are chronicles, composed of characteristic events and temporal constraints that events have to satisfy.

The knowledge base embeds about 25,000 objects corresponding to 33 goals, 27 tasks, 75 inference structure, 3,200 concepts and 2,000 relations (in KADS sense). The knowledge acquisition was a hard work: it has needed 6 knowledge engineers and up to 13 experts. Together they produced a common glossary about the process control, a General Expert Analysis (it required one year) and a Detailed Expert Analysis (2 years). The expertise exists under tree ways: reformulations (natural and structured language) able to be validated by the experts, a model on OpenKADS for the software engineers and, at last, Kool codes. The total cost of the knowledge acquisition is about 30 man-years (14 for the preliminary knowledge acquisition, the rest for the tuning and validation phase). It represents about 100,000 lines of Kool code.

5.6.5. Output

The output of Sachem mainly consists in the presentation of the BFP to the Blast Furnace operators. The recommendation module is being developed. On the screen we mainly have:

- a synthesis of the process control with the presentation of the alarms and warnings (the BFP considered as important),
- the possibility for the operator to ask for the justifications of the BFP,
- the possibility to consult the process data,
- the summary for the next shift,
- the prediction of the cast iron temperature.

In the next version we will find the recommendations of actions.
5.6.6. Implementation
The architecture of Sachem is presented in figure 8. Data are computed in batch mode every minute. The total cycle duration is of 8 s on a SP2 (a computer of the range of Deep Blue). The total cost of the project is about 200 man-years. The project began in 1991.

5.6.7. Results
The expected gains are of 6 francs per ton of iron. The production in Sollac is of 10 millions of tons per year. The Sachem system is in operation on 3 BFs out of 7. The first one was provided in October 1996, the second in May 1997 - both at Fos sur Mer - and the third in September 1997 at Dunkirk. The first observations show that the number of process anomalies is divided by 3. There are even some problems that have completely disappeared since Sachem is in service.

5.7. Exxon: TIGER project
The TIGER condition monitoring system was devised and implemented within the framework of the TIGER Esprit European project entitled: “Real Time Assessment of Dynamic, Hard to Measure Systems”. It was applied to the gas turbine application domain [3, 4, 6]. The monitoring functionalities of TIGER are provided by three independent tools which can all work in parallel, each examining different aspects of the turbine:

1. Kheops [43] is an expert systems shell allowing one to compile the rule base into a decision tree, which guarantees an upper response time limit. Kheops is used as a hight speed limit checking system.

2. IxTeT [25] is a temporal reasoning system which is able to perform on line recognition of chronicles, that is sequences of events related with temporal constraints. The chronicles are specified by the user for normal operating conditions and/or known faulty situations.

3. Ca~En is a model based detection and diagnosis system. It uses deep models –physical laws of the domain-- of the physical system normal behavior and implements a consistency based reasoning schema. Fault knowledge is not necessary which makes Ca~En usable from the very beginning of the physical system life.

This section focuses on the Ca~En tool applied to the Exxon turbine which is one of the two applications dealt with during the TIGER project (the reader can refer to the section 5.8 for a description of the second application).

5.7.1. Monitored physical system
The initial installation of TIGER was at the Fife Ethylene Plant which is a 650,000 tones a year gas craking facility located in South East Scotland, and jointly owned by Exxon Chemical company and Shell Chemical company. The major product of the facility is high grade ethylene for use in the plastics and butyl rubber industries in both the UK and on the continent. The feed stock is ethane gas obtained from the Shell/Esso offshore facilities in the North Sea. The process is continuous with the ethylene product being transported by both ship and pipeline to end users in the UK and on the continent. The gas turbine is a 28 mega-watt General Electric Frame Five with two shafts supplied by John Brown Engineering (see figure 9), this is used to drive the primary compressors for the fife ethylene plant.

The turbine is controlled by a Speedtronic Mark IV controller which is in charge to apply all the regulations which are necessary to satisfy the production demand while keeping the efficiency optimal. The Exxon turbine is a two shaft turbine with a set of nozzles, known as “second stage nozzles” that balance the energy between the two shafts. This allows the compressor to run at its optimum speed (5100 ± 10rpm) while providing for variable load on the turbine. The position of the second stage nozzle is controlled by a servo. The decomposition into subsystems is visible on figure 9.

Two subsystems were deeply analyzed: the second stage nozzles and the fuel admission system.
5.7.1.1. Operating modes. As the Exxon application turbine runs continuously 24 hours a day, the starting and stopping modes are very rare and the most interesting operating mode is the nominal speed mode. However, within the nominal mode, still two possible modes exist:

- The speed control mode takes over when the exhaust gas temperature is below the acceptable limit. The admission control law is hence a function of the load which increases or decreases the primary shaft rotation speed;
- The temperature control mode takes over as soon as the exhaust gas temperature reaches the unacceptable limit. Increasing fuel admission is not allowed anymore. If more power is needed though, the steam from an auxiliary steam turbine can be increased. It is also possible to add steam right into the combustion chamber manually.

Besides, the turbine can be fuel feeded from two sources: starting fuel tank and running fuel tank. This also defines two distinct operation modes. Each source is equipped by two valves: the first valve keeps the input pressure of the second constant whereas the second valve controls the injected fuel flow.

5.7.2. Monitoring aims

The Exxon application turbine needs to be monitored mainly once it is established around the nominal speed. The requirements are hence on-line monitoring with real-time constraints. These three following aspects must be present:

1. Assistance for controlling the turbine which has the following requirements:

- to detect abnormalities and to produce a simple to interpret and precise report about the state of the turbine to be used by the operators.
- to interpret the alarms coming from the control system ladder logic; these may be ambiguous as the same alarm may have several possible causes.

2. Maintenance which has the following requirements:

- to go back to the primary causes of detected abnormalities and to produce a diagnosis (list of possible causes in terms of responsible faulty components).
- to perform trend analysis on the basis of one month or one year scale recorded data; this is intended to detect deteriorations resulting from wearing.

3. Anticipation since the faults must be detected as soon as possible in order to allow operators to fix the problems without stopping the turbine.

5.7.3. Inputs

The sensors existing on the turbine provide 74 analog signals and 80 digital signals. Most of the continuous signals directly provide the value of a given physical quantity of the turbine through time. Still some of them result from a simple calculus (percentage for example). They are generally noisy. These signals were first filtered (low band filter) before being used as inputs for the TIGER system. Digital signals are representative of the state of the turbine. They constitute a set of alarms which must be interpreted by the operator when deciding upon the control actions to be performed. This is a difficult task for which the manuals containing the logical circuit drawings are often necessary to consult; indeed, the alarms are not one-to-one associated with the faults. All these sensors primarily arise from control requirements.

5.7.4. Models

Ca~En’s representation formalism allows one to combine empirical causal knowledge and first principles of the domain. Time is dealt with explicitly with a logical clock which delivers a constant frequency sampled version of continuous time. The Ca~En formalism is based on a multi-model representation scheme including:
– a causal model in the form of an influence graph (see section 4.2) in which the links represent influence relations between pairs of variables, also called the local constraint level;
– an analytical equation model which allows one to represent algebro-differential equations, also called the global constraint level.

Both models can manage imprecise knowledge. They take part to the prediction algorithm in a cooperative cycle. The prediction process being driven by the causal model, this requires that all the knowledge about the physical system is implemented at this level. Consistency of the pieces of knowledge which are implemented at both levels is guaranteed by the fact that the causal model is generated from the corresponding equational model automatically. This is performed by the Causalito algorithm [44] which implements a causal ordering approach for multi-model systems.

Ca~En has two processing modules:

1. A simulation module which produces the explicit behaviour of the physical system in terms of the values of the internal variables across time according to the behaviour of the exogenous variables. Imprecision is managed with interval values, which implies that predicted graphs are curve envelopes [45, 46].

2. A diagnosis module which accounts for fault detection and isolation of faulty components. Fault detection is based on models of normal behaviour. The on-line simulation of these models provides a way of implementing a discrepancy detection procedure. This enables monitoring of the behaviour of the system and detection of early deviations from the nominal behaviour. The diagnosis algorithm falls into Reiter’s model-based diagnosis framework and uses the Ca~En causal graph as the System Description (SD). It relies on the collection of conflict sets, i.e. sets of components such that the observations indicate that at least one of the components in a set must be behaving abnormally, and the use of an incremental hitting set algorithm. The diagnoses are given as sets of faulty components labelled by their corresponding time of failure [47].

The Ca~En diagnosis system conclusions rely on a reasoning based on the physics underlying the behaviour of the system, i.e. physical laws and empirically known causal interactions. When a fault is detected, it is viewed as the violation of some of these physical principles which then guide the isolation of the faulty component(s). As a consequence, there is no need to anticipate the faults, which is highly valuable in most complex engineering domains. On the other hand, as variables and parameters take interval values, one can easily adapt the models granularity to the requirements of the faults. Hence Ca~En has a wide coverage of faults, from those radically changing the behaviour of the physical system to those causing smooth deviations.

5.7.4.1. Behavioral models. There are two sources of knowledge:

– the Speedtronic Mark IV controller manual which provided us with the relations between the control and controlled variables in the form of a set of equations. Some of the parameter values were also available;
– data recorded on the turbine which allowed us to identify the structure and parameter values of other relations.

5.7.5. Outputs

The outputs are, for every non exogenous variable, the curve envelop providing a bounding of their possible values across time. Moreover, an anomaly detection message is displayed every time a variable is considered to have an abnormal behavior. This message is followed by a diagnosis report listing the diagnosis sets for which the components are labelled by their failure time. A screen display is given in figure 10.
5.7.6. Implementation

The Ca~/En system used in TIGER was implemented on a Sun 4 station, written in LeLisp. A more recent version runs on Solaris and includes a component-connection knowledge acquisition interface.

5.7.7. Validation, maintenance, evolution

The behavioral models were fitted to more than ten recorded data scenarios chosen for the various faulty situations they covered. Validation was performed by running Ca~/En on one hundred scenarios chosen randomly from one year recorded data. The diagnosis results were found correct for 99 scenarios out of 100.

5.8. Dassault Aviation : TIGER Project

The TIGER condition monitoring system, as described in the anterior section, was used to develop two applications during the TIGER Esprit project. The anterior section presents the application made to the Exxon turbine; this section presents the application made to the Dassault Auxiliary Power Unit turbine.

5.8.1. Monitored physical system

The APU (Auxiliary Power Unit) is a little turbine used as an auxiliary power supply in aircrafts. The one that was considered was a 0.4 MW turbine designed by the company Micro Turbo for Dassault Aviation and used in Rafale fighters. Like all turbine systems, it is made of an air supply, a compressor, a combustion chamber, a turbine and an exhaust pipe. It is used on the ground or during flight time to produce electric or pneumatic power:

- electric power: the APU is used to drive one or two electrical power generators.
- pneumatic power: this is generally performed by taking air right after the low pressure compressor to realize the following functions:
  * starting the jet engines in some faulty situations;
  * cabin air conditioning, particularly at take off time when the engines need full power;
  * defrosting.

The TIGER application focused on the APU fuel system which feeds and regulates the APU, providing the fuel from the aeroplane tanks to the injectors with the right pressure and flow, depending on the shaft speed and the aeroplane operating mode. This subsystem is given in figure 11.

The APU fuel system includes the following functions (see figure 11):

(i) A pressure and flow rising function including a fuel shut-off valve which opens and closes the fuel system; a check-valve which enables to fill the circuit with fuel at starting time, when the pump has started the derived flow closes immediately the check-valve; a pump which provides desired flow and pressure, the pump is a volumetric pump with constant capacity, its speed is proportional to the engine speed.

(ii) A filtering function constituted by three filters located just before the fragile components, e.g. the fuel control valve. Their role is to eliminate the impurities (dust, ice-crystals, etc.) that can be present in the fuel.

(iii) A fuel regulation function constituted by a fuel control valve associated with a current servo which regulates the fuel flow as a function of the APU operating mode and of the running speed set point; a differential pressure control valve which maintains constant pressure between the fuel control valve input and output so that there is a proportionality relation between the servo-current and the subsection of the valve, hence the fuel flow. The exceeding fuel arriving at the input of the fuel control valve is recycled just before the pump.

(iv) A fuel injection function which includes two injection rings which spray the fuel in the combustion chamber; the first injection ring of four injectors acts on its own for low power supplies, it is complemented by the second ring of five injectors when high power is needed. The second ring is activated as soon as the dividing valve opens, which is obtained for some pressure conditions.

(v) A drainage function which avoids, when the APU is stopped, to have accumulated fuel in the circuit, this being dangerous in case of too high temperature situations. It is composed of a drainage valve which empties the fuel out of the secondary injection ring, a second valve which has the same function for the first injection ring and a shut-off valve which opens when the APU is stopped.

The APU dynamics is very quick. The sensors sampling time is 0.02 seconds.

5.8.1.1. Operating modes. The functional description of the APU fuel system shows that some components are of continuous type and others, which control the opening and closing of some parts of the circuit, are of “binary” type. The binary components define 8 different operating modes associated with pressure conditions.
5.8.2. Monitoring aims

The function of the APU is to assist the main engines in situations which require full power. If a problem occurs, the starting phase is at least as important as the running phase to detect and diagnose the fault. It is often the case indeed that a problem results in the APU not starting at all. Therefore, the APU monitoring must be particularly efficient for transients, i.e., during starting time. Even though the problems need to be detected as soon as possible, on-line monitoring is not required; the data can as well be analyzed a posteriori. Therefore, monitoring exclusively relates to the maintenance aspect, as an assistance provided to the operator when the turbine is on its test bench after an anomaly, or for a series of programmed maintenance tests.

5.8.3. Inputs

Given the type of monitoring which is required, the available sensors are those existing on the test bench, whose number is higher than the ones set on the turbine during normal operation. There are 12 test bench sensors. They all provide continuous signals, each reporting the value of a physical quantity of the fuel system across time.

5.8.4. Behavioral models

The knowledge was provided on the one hand by the Dassault Aviation experts which specified the APU, and on the other hand by the Micro Turbo experts which designed the APU. The knowledge was mostly in the form of equations coming from the physical domain; there were also numeric knowledge about the parameter values and their tolerances as well as graphs obtained on the test bench for characterizing the APU efficiency. The fuel system was decomposed down to 10 hydraulic components of 5 different classes. A generic model was built for every class. The global fuel system model includes 22 equations for 4 input variables, 18 internal variables and 22 constant parameters. 8 variables were measured and 14 were not.

5.8.5. Validation, maintenance, evolution

The behavioral models as well as the CaEn diagnosis system were tested on 25 scenarios recorded on the Micro Turbo hydraulic test bench. The scenarios included several faulty situations obtained by physically introducing a fault into the system. A demonstration of CaEn running on-line on the test bench was performed and resulted fully successful.

5.9. France Telecom : IMOGENE project

IMOGENE (Inversion of a Model thanks to Genetic algorithms)\(^4\) is a supervision system whose main function is to determine, in a given network, which streams are responsible for call losses on communication organs (switches and circuit groups).

IMOGENE performs this task by comparing stream traffic values to their nominal values. However, stream traffic values are not measured by the on-line data acquisition system and, hence, have to be computed. We perform this computation by inverting a stream propagation model thanks to evolutionary computation techniques [48].

5.9.1. The French long distance network

In the first hand, the French long distance network can easily be seen as a graph with:

- nodes, representing commutation centres (or switches) whose capacity represents the maximum number of simultaneous carried calls,
- arcs, that represent the circuit groups between two switches; their capacity (i.e., their number of circuits) is equal to the maximum number of calls that can be carried simultaneously.

Then, the telephone network carries some streams, each stream corresponding to a set of calls going form a switch (called the origin node) to another one, the destination node; a stream is also characterized by its of-

\(^4\)collaboration between France Telecom/CNET and LAAS/CNRS
ferred traffic value (the average number of call arrivals during a period corresponding to the average call holding time) and its routing table that provides the possible ways to reach the destination node.

The basic node of this network is the subscriber switch, called the routing autonomy switch (RAS), which is connected to phone subscribers, PS, either directly, thanks to point-to-point links, or thanks to a local switch (which might be a satellite), marked LS in figure 12.

The network has a specified structure in which some nodes are only dedicated to calls routing and are not directly connected to the phone subscribers: the transit switches. The French long distance network is composed by two levels: the main transit level (formed of MTS) and the secondary transit one (where the switches are marked STS), as shown in figure 12. In this way,

![Fig. 12. Telephone network hierarchy](image)

when a subscriber A tries to call a subscriber B who is not connected to the same routing autonomy switch, the communication request of A tries to go through the high usage switches and circuit groups, as defined in the request stream routing table. If one of these circuit groups is overloaded, the call request of A overflows, as written in the routing table, toward a final circuit group and it goes through the route corresponding to this overflow.

The routing policy of the French long distance network only allows one overflow: every call that tries to overflow towards an overloaded circuit group is lost.

5.9.2. Supervision objectives

Independently of the network switches and circuit groups characteristics, the traffic flow quality can be affected by some streams overloads and, because of the high connectivity of the network, this disturbance may propagate within the network in a very short time frame, creating new disturbances.

This traffic modification can be linked to:

- daily or seasonal changes: some switches offered traffic increase whereas the others decrease. This is, for instance, the case of switches connected to tourist sites during holidays.
- a whole overload: this may happen on special occasions like Christmas or New Year’s day.
- an unexpected event such as a televised or radiotelephone game, a natural, rail or air disaster or else a political event.

The real time network control objective is to avoid network degradations by supervision means and by the implementation of traffic control actions; these actions permit, in the case of an overload, that most of the call requests reach their destination, using the maximal number of available resources in the network.

The real time network control can be decomposed in the following way:

1. network state and working conditions supervision,
2. supervision data collection and analysis,
3. network abnormal circumstances detection,
4. disturbances causes diagnosis,
5. corrective actions over the network or the traffic.

5.9.3. Inputs

IMOGENE inputs come from a data collection that is done every 5 minutes on the various network elements (switches and circuit groups). These data principally concern the call attempts number, the effective calls number and the offered and carried traffic.

IMOGENE knows the network structure, i.e. the switches number and capacity, the circuit groups number and capacity and the streams routing table.

5.9.4. Model

The model used by IMOGENE is a one-moment model (i.e. offered traffic is only characterized by its mean) and has been developed under classical assumptions of this kind of model:

- A1 Call arrival is a Poisson process.
- A2 Call holding time has a negative exponential distribution.
- A3 Blocking probabilities are statistically independent.
– A4 The network is in statistical equilibrium.
– A5 Call arrival (node-originated plus overflow) on any element is a Poisson process.

The inputs of the stream propagation model presented here are the same as the ones of the CNET© simulator SuperMac, i.e. the network structure and the streams offered traffic values. On the other hand, our model is more informative than SuperMac concerning its outputs since as it provides not only the whole traffic losses and offered traffic for each organ and for each stream but also the traffic losses and offered traffic for each stream at each network element.

Our stream propagation model is qualitative in that it is only based on [49, 50]:
– the intuitive notion of blocking organ which is based on the use of the Erlang’s formula; our model, at the opposite of the models described in [51–53], uses this formula as an indicator and not as an equation that has to be solved.
– a qualitative knowledge about the network, based on its structure.

5.9.5. Outputs
IMOGENE outputs are the approached values of the network stream offered traffic; these values are computed by inverting our stream propagation model thanks to evolutionary computation techniques.

Then, these values are compared to the corresponding nominal traffic values in order to determine which streams are responsible for the overflow.

5.9.6. Implementation
IMOGENE has been elaborated as part of a CNET-CNRS collaboration contract and has been implemented in C++ language, on a Solaris UNIX station.

5.9.7. Validation, maintenance and evolution
IMOGENE has been tested on a set of networks that present all the particularities of the real one; it gives particularly good results when using a real variant of a genetic algorithm.

On going work deals with the complexity and real-time constraint requirements in the following aspects:
– add to IMOGENE a qualitative pretreatment stage allowing us to decouple the network in several independant subnetworks.

– consider parallel evolutionary computation technique.

Moreover, we are working on the following issues:
1. control actions that bring back the network to normal operating conditions,
2. mobile phones and the disturbances they cause on the French long distance network.

6. Synthesis and Conclusion

6.1. Synthetic tables

In the next three tables, we compare the applications the members of the “ALARM” group are involved in, according to criteria which were appreciated as significative ones. The first table (see Figure 13) compares the applications with regards to their context: which kind of observables can be obtained? what is the dynamics of the monitored system? what is the general goal of the application: preventive maintenance or control of the system. The second table (see Figure 14) compare the applications in a more detailed way. Which tasks do they complete: filtering alarms, detecting malfunctions, locating abnormal components, looking for primary causes of malfunctions, proposing (control or repair) actions? On which kind of models do they rely: numerical, qualitative? static or dynamic? expertise-based? Which artificial intelligence techniques do they use: influence graphs, causal graphs, rule-based systems, pattern recognition or anything else? Which kinds of reasoning schemes are implemented: deductive, abductive or another kind?

6.2. Conclusion

This paper provides a summary of the different techniques that are nowadays proposed for alarm monitoring of dynamic systems in the field of Artificial Intelligence together with the description of various applications implementing these techniques. The variety of theoretical approaches used, as well as the good deal of different real-world problems tackled with success, demonstrates that AI techniques offer operational solutions to industrial problems involving these issues. The applications and corresponding monitoring systems are analysed with respect to different criteria which exhibit a possible taxonomy of the domain. This work is to be related to other works in the same direction by Chantler [54] and Console [55].
The results of our study show that the very first feature orienting the choice of the technique is related to the continuous or discrete nature of the system. The GASPAR application dealing with the supervision of telecommunication networks is definitively discrete whereas the TIGER or DIAPASON applications deal with continuous processes.

However, one should note that the discrete/continuous feature is not intimately linked to the process at hand. It also depends on the level of abstraction at which the process is observed. Indeed, the IMOGENE application deals with the supervision of the telephone network by aggregating the calls, that are discrete quantities, in terms of traffics and flows. On the other hand, the DIAPRO application reasons from an interpretation of the continuous signals in terms of discrete events. The choice of the level of abstraction is very much related to the type of knowledge available and its source. The actual tendency is towards the increasing use of models from the design stage as this knowledge is generally accessible in a well formalized form and can be transformed, often in an automatised manner, as desired.

As a matter of fact, the so-called model based approach is nowadays taking importance over the more classical shallow knowledge approach in the dynamic systems monitoring area.

7. Acknowledgments

The following persons also contributed to the group’s work that gave rise to this paper: M. Allouche (SIMADE), S. Bibas (LIPN), P. Dague (LIPN), G. Deflandre (France Telecom), M. Dumas (CEA), D. Fontaine (UTC), S. Gentil (ENSIEG-LAG), G. Ramaux (UTC), L. Rozé (IRISA), C. Sayettat (SIMADE).

References

### Objectives

<table>
<thead>
<tr>
<th></th>
<th>Alexip</th>
<th>Austral</th>
<th>Diapason</th>
<th>Diapo</th>
<th>Imogène</th>
<th>Gaspar</th>
<th>Retrait.</th>
<th>Sachem</th>
<th>SOLLAC</th>
<th>EXXON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abnormalities Detection</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alarm Filtering</td>
<td>*</td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Situations Identification</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fault Localisation</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary Cause Determ.</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action Advice</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>in progress</td>
<td>-</td>
<td></td>
<td>-</td>
<td>future</td>
<td>-</td>
</tr>
<tr>
<td>Prevent/Correct Action</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>P</td>
<td>C</td>
<td></td>
<td></td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

### Physical System Knowledge

<table>
<thead>
<tr>
<th></th>
<th>Alexip</th>
<th>Austral</th>
<th>Diapason</th>
<th>Diapo</th>
<th>Imogène</th>
<th>Gaspar</th>
<th>Retrait.</th>
<th>Sachem</th>
<th>SOLLAC</th>
<th>EXXON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Numerical Model</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Numer. Model</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>partial</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete Events Model</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualitative Model</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Tools and Techniques

<table>
<thead>
<tr>
<th></th>
<th>Alexip</th>
<th>Austral</th>
<th>Diapason</th>
<th>Diapo</th>
<th>Imogène</th>
<th>Gaspar</th>
<th>Retrait.</th>
<th>Sachem</th>
<th>SOLLAC</th>
<th>EXXON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Influence Graphs</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Causal Graphs</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td></td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rules-based Systems</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td></td>
<td>(old version)</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenarios recognition</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>AMDE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others…</td>
<td>decision trees</td>
<td>action plans, automata</td>
<td>situations prototypes</td>
<td>genetic algs</td>
<td>automata</td>
<td>numerical equations</td>
<td>mathematical models</td>
<td>numerical constraints</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Use

<table>
<thead>
<tr>
<th></th>
<th>Alexip</th>
<th>Austral</th>
<th>Diapason</th>
<th>Diapo</th>
<th>Imogène</th>
<th>Gaspar</th>
<th>Retrait.</th>
<th>Sachem</th>
<th>SOLLAC</th>
<th>EXXON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Explanation</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>


béry, France.


shop on Principles of Diagnosis (DX ’94), 1994.


[53] J. Guérineau and J. Labetoulle, “End to end blocking in tele

tional Workshop on Principles of Diagnosis (DX ’98), (Cape Cod, Massachusetts,)), pp. 6–15, 1998.