Supervision applied to nuclear fuel reprocessing

Jacky Montmain

Supervision Applied to Nuclear Fuel Reprocessing

Jacky Montmain
LGI2P, URC EMA-CEA
Site EERIE
30035 Nmes Cedex 1 (*France*)
Phone : (*+33*) *4 66 38 70 58*
Fax : (*+33*) *4 66 38 70 74*
jacky.montmain@site-eerie.ema.fr

Model-based supervision developed by systems analysts has become an acknowledged supervision aid, ensuring early detection of malfunctions and thereby allowing control of the availability and vulnerability of a process facility. However, it is associated with diagnostics of the process itself, and not of the process control situation, which is the veritable subject of supervision. The operator, facility, control triplet determines a complex situation that must be considered from multiple viewpoints beyond knowledge of the single behavioral model usually advocated in process control approaches. Representing different aspects of process control situation from multiple viewpoints notably allows the on line selection of the behavioral models relevant to the observed situation. Given the size of the application, it was essential not only to structure the knowledge required for the supervision system functions into operating system viewpoints, but also to provide a unique representation method for each viewpoint. The systemic approach SAGACE provides this formal representation framework and the methodology adopted to design and implement our industrial prototype relies on it. All these principles are illustrated by a description of an industrial application in the area of nuclear fuel reprocessing : the size and complexity of the facilities and their high degree of computerization make reprocessing particularly well suited for supervision applications.

Keywords: Supervision, Diagnostic reasoning, Technical processes, Nuclear fuel reprocessing.

1. Introduction

The productivity of an industrial facility can be improved by automating the means of production. ”Automation” implies that the actions performed to obtain the product are implemented systematically and consistently, with controlled variance of the characteristic product parameters. When suitably implemented, automation should result in a virtually ideal solution for a fabrication process. An optimization strategy involves two major steps : creating a mathematical model of the facility, then calculating the control signal that must be applied to the model to meet a productivity criterion. The first expected result is thus a higher quality finished product or, for the same quality level, an economic gain resulting from rationalization of the technical specifications at a less costly level [25] [2].
This theoretical situation is rarely achieved in practice, however, as it implicitly assumes that the elementary process, sensor, actuator and regulator functions are correctly implemented. Moreover, systematic automation can have insidious side effects, seriously impairing the production system. Hence the addition of a supervision level to assess the process state at all times, supply validated data to the process control system, and thus improve process availability and maintainability.
These considerations explain the rapid development of plant supervision and the success encountered by process diagnostic techniques in recent years. However, although considerable work has been published concerning model-based diagnostic methods, fault detection and isolation (FDI) algorithms applied to textbook examples and simple applications or applications of limited scope, very few papers address the suitability of such methods for complex industrial-scale applications. In the survey proposed by [12], it is surprising to note that only 25 of the 166 examples reviewed and analyzed involve systems with more than five faults, and only two actually concern a complex process (24 faults in a ship diesel engine, and 30 faults in a nuclear reactor coolant pump). In the light of this recent analysis, much work apparently remains to be done before the methods proposed in the literature actually meet the general objective described above, and can be applied to industrial processes.
There are a number of reasons for this situation, including the inadequacy of available industrial models for

AI Communications 0 () 0
ISSN 0921-7126 / $8.00 © , IOS Press

the proposed FDI algorithms (notably in Process Control Theory), the multiplicity and difficulties of compiling the knowledge required for plant diagnosis, and instrumentation initially designed for anything but supervision [20]. For example, two types of models are generally available in industrial plants : material or energy balances established from process block diagrams and flowsheets integrating operator knowledge of production rules ; and complex, partial derivative and nonlinear analytical equations written by physicists or chemists when the focus is the process and the physical phenomena involved. In large plants, the former are written from a production management standpoint and thus implement shop-scale balances, while the latter are developed to obtain load diagrams or for training simulators. In any case, they were conceived for purposes other than supervision, and classic FDI techniques used in automatic control (generalized parity space or batteries of observers [7] [10] [22]) are poorly suited to this type of representation. Automatic control models are local control models, and are thus not adapted to the supervision of a complete facility [23]. In an attempt to surmount these shortcomings, new techniques and notably artificial intelligence have been adopted in recent years [6] [8] [9] [11] [17] [18] [21] [27].

Even if this aspect is all too often illustrated by applications of restricted scope for which convoluted solutions are sometimes necessary to coax the models into a diagnostic role, the other stumbling block which is not technical one and is much less often discussed in the literature concerns the very purpose of the supervisory system in the control room.

Until now, availability monitoring has often been neglected because the measurement consistency analysis and fault detection functions are implicitly performed by the human operator, and so generally overlooked or only partially incorporated in the automation scheme. It would then seem reasonable to postulate that the objective of supervision is to automate decision-making tasks. The nature of the information supplied by a supervisory system appears to corroborate this point of view : for automatic control specialists, the information supplied by a supervisory system is based on the use of mathematical process behavior models, which in turn are generally based on classic state-variable representations. This information would be difficult to use by a control operator, not only because it corresponds to a representation of the facility with which he is certainly not familiar, but also because it is not necessarily pertinent in an operating context. It is in fact intended for the Supervisory Control and Data Acquisition system (SCADA) rather than for the operator, who is removed from the control loop. The supervisory system provides validated results to the SCADA, which reconfigures and computes new control laws in any degraded operating mode detected. It is the active supervision perspective. The objective is not to help the operator in a critical situation to analyze a process that computerization has made increasingly complex, but rather to discharge the operator of decision-making tasks in a fault situation.

In addition to the social problems it could well raise, this option is based on mathematical knowledge that is assumed to be exhaustive. But how to be sure that every malfunction has known consequences whatever the current operating situation ? How to ensure that the corrective or remedial action required for a given malfunction are the same regardless of the operating context or strategy ? FDI approaches basically integrate the notion of behavior in their supervisory models, and these aspects related to operating principles are largely disregarded ; how then to construct a system comprehensive enough that it can do without human faculties of adaptation ? All these questions weigh heavily on the acceptability of such supervisory systems in a control room context, and are perhaps the veritable reason behind the limited application of FDI algorithms.

Many other dimensions must be taken into account before the surveillance of process physical parameters can be extended to the supervision of plant facilities. The action necessary in the event of a malfunction in complex large-scale processes will thus remain subject to the decision of the operating staff for many years to come. Should the objective of supervision then not be revised ? Would it not be more reasonable to envisage systems that no longer eliminate the operators from process control, but instead aid them in this function by assisting their reasoning in the current situation [26] ? Supervision understood in this way (to which the disparaging term passive could be applied in opposition to the preceding option) is a no less ambitious project : assimilating the human dimension in the system integrates an additional level of complexity. A process behavior model alone is no longer sufficient ; what is required is a control situation model in which the operator is an essential constituent. The application discussed here is based on this second option.

2. Diagnosis and Operator

Targeting the supervision system on the operating team rather than the SCADA leads to major revisions

in model-based diagnosis strategies [20]. The intelligibility and pertinence to the operator of the results provided by a diagnostic system become legitimate issues. The results of the diagnostic system become part of the reasoning of the operator analyzing the situation, and as such must be substantiated and explained. This terminology is not unlike that of expert system or qualitative physical approaches [14] [15] [18].

However, a number of difficulties are raised even by attempting to integrate the concept of early fault detection (the foundation of FDI algorithms) into the context of classic process control and operating principles. Indeed it may be noted that in many large industrial plants the operators benefit from an appreciable control margin with respect to the nominal operating parameters. Consequently it may not necessarily be considered significant to inform the operator that a component has deviated from the standard operating criteria as long as it meets its input/output function requirement. From an operational standpoint, when the setpoint and output of a regulated system fully coincide, there is no veritable reason to check in addition that the actuator is operating consistently with its prior calibration setting. The actuator position becomes an issue only in the event of a discrepancy between the setpoint and the output ; otherwise it is considered to be of no interest. The state of the process does not have the same meaning to a control room operator and to an automatic control specialist in dynamic systems.

In conventional process control theory, ensuring that the process is operating correctly in terms of input/output does not prove that the system is not in a degraded operating mode : the current mode may be considered ”nominal” or ”normal” with respect to its efficiency but, depending on its internal state, may be more sensitive to further perturbation or malfunctioning. The control margin is thus restricted without affecting current operation ; only the onset of a new disturbing event or a change in the setpoint will reveal the deviation between the assumed normal state and the actual state of the system. This eventuality corresponds to the system vulnerability ; failure to control vulnerability can lead to crippled-mode operation or to costly and laborious shutdowns. The concept of vulnerability is easily understood by an operating team, and is also an excellent means of emphasizing the importance of early fault detection in the control room.

Consider an example from the field of nuclear fuel reprocessing, discussed in Section 4. A drop in the level of the uranium front in the extraction column may be detected by a variation in the column weight differential. The problem of stabilizing the uranium front is a symptom that should not affect the process if the operating conditions remain unchanged. The only consequence of this anomaly is thus a restricted control margin : the uranium descends lower in the column. Only the system vulnerability is affected. Conversely, if other control parameters were modified, the reduced margin could become critical : the eventual consequence would be uranium leakage from the base of the column, which would require recycling and result in a loss of availability. In our application, we chose whenever necessary to assign two detection methods to each malfunction : early detection corresponding to symptoms that do not necessarily have any observable consequences on the process outputs (anomaly detection), and later detection based on abnormal variations in the observable characteristic operating parameter values (fault diagnosis). These two functions are covered in detail in Section 4.

When this chronological sequence in the scale of gravity of dysfunctional events is clearly perceived by the operator, the related principle of early fault detection can more easily be accepted in the control room. The notion of early detection is thus incorporated into a control situation representation system shared by all the operators. It is merely an example illustrating that all the supervisor results must be processed and output with the same concern because they are addressed to the human operator. The context of this collective representation is considered in further detail in the following section 3.

3. Cognitive Automation or Operator Assistance

Before presenting our model representing the knowledge required for supervision, it is important to distinguish between cognitive automation and operator assistance. The introduction of an operator aid system in the control room inevitably has an impact on process control. A human operator’s action is based on representations constructed by the ”specialized artifact” : i.e. the ”physical symbolic system” constituted by the control desk. Any attempt to modify the operator’s workstation, for example by adding advanced features, thus unavoidably modifies the process control situation itself as the operator ”intelligently” adapts to the proposed new representations.

3.1. Cognitive Automation

In the case of a complex industrial process (notably in hazardous industrial situations such as found in the energy, transport, defense and aerospace industries) the operators must control highly automated facilities. Automation of monitoring provisions and some actions [3] may be further extended to "cognitive automation", i.e. replacement of the operator for certain decision-making functions. "Today it is increasingly clear that, contrary to expectations, automated monitoring provisions can make the operators' tasks more difficult ; the same may be true of cognitive automation" [1]. Under these conditions, the operator intervenes only in situations not anticipated in the automated system design, situations where the unexpected becomes possible, and where only human intelligence can cope. In this hypothesis, the automated system reverts to operator control when the process is degraded and only safety-related functions are ensured. The operators must then recover the situation by taking account not only of the process functions, but also of prior decisions made by the automated systems which further complicates their task (Bainbridge refers to this phenomenon as the paradox of automation).
Degraded-mode operation in automated contexts relies to a large extent on implicit reasoning, leading operators to pool their points of view and construct a common operational frame of reference [4]. Under these conditions, it is not a trivial matter to add artificial agents to the team in the form of operator aid systems [16].

3.2. Process Control Situation

The decision to intervene in the situation depends on the meaning (operational knowledge) and value (assessment of the utility of the information) that the observer accords to the perceived phenomena. Under these conditions, it is less a question of explaining the situation by identifying cause-and-effect relations than of "refocusing", i.e. constructing a suitable representation for action.
To allow intervention at the selected level of reality by the observer faced with difficulties of understanding, anticipating or controlling it, the situation must be given form and meaning through representations (decision-making aid). In fact, the observers may be considered to see themselves from an external viewpoint confronted with a knowledge object (the process control situation) about which they construct a representation consistent with their proposed intervention. In attempting to act on a remote situation, the observer is in a position of dynamic action, imagination and decision-making.
The supervision of a facility controlled by an operating staff defines a complex situation that an observer (an operator or a supervision system) attempts to diagnose. In the first eventuality, the operators analyze their own conduct, while the second involves an external observer. The supervisory system aid provisions must therefore address the operator, apprehend the process and take control actions into account. Thus the supervisor (an observer but not an operator) must also be provided with a process control representation.

3.3. A unified representation of the diagnosed situation

FDI techniques developed by systems analysts are associated with diagnostics of the process itself, and not of the process control situation, which is however the veritable subject of supervision. The operator, facility, control triplet determines a complex situation that must be considered from multiple viewpoints beyond knowledge of the single behavioral models to achieve a relevant on-line diagnosis.
As previously mentioned, model-driven diagnosis traditionally calls on a process behavior model : differential algebraic equations based on the fundamental principles of the process, but which integrate little or no representation of the process control situation. However, interpreting the control situation calls on other knowledge than behavioral knowledge alone. The models describing the behavior of the facility according to the current process control situation must be selected or revised to ensure the diagnostic is pertinent at each instant. The problem is currently addressed by the use of hybrid models : process control events (changes in the objective, the operating mode, the configuration or the operating strategy) are managed by systems consisting of discreet events (e.g. a Petri network) that can use the results of numeric behavioral models or manage their coherence depending on the situation. However, either these systems take into account only a single aspect of process control (a single type of event, such as the operating configurations) and thus provides a reduced control situation model, or several types of events are indeed taken into account, but the lack of data source homogeneity is liable to diminish the coherence or the pertinence of the model.
The variety of these process control situation events re-

quires a precise classification. All of them are not relevant to a single viewpoint. The SAGACE method developed in our laboratory is based on a systems analysis approach ; it permits such a classification and allows us to obtain a representation of the operating situation. In SAGACE, the situation may be considered from a functional, organic or operational standpoint. The functional view is an external view of the phenomenon as a system open to its environment. The organic (or ontological) view is an internal view of the system as a network of interrelations and interactions among operative, logistic and auxiliary components. The operational (or teleological) view seeks to clarify the decision-making competencies involved in order to accomplish the objective (control and management). A more refined typology may be obtained by combining the three views from the perspective of examining certain expected system properties : performance, stability and integrity. The combination of the three views and three perspectives determines nine system viewpoints identified in the SAGACE matrix [24]. Each of these viewpoints can be used to select the pertinent behavior model on line. This ensures that the operator aid system diagnostic result is consistent with the process control situation.

The knowledge representation of the system to be supervised by the SAGACE method involves a projection on the nine-viewpoint matrix to assess the complexity of the system by distributing the knowledge and questions over the viewpoints. The generic matrix is shown below in Figure 1. Each viewpoint in the matrix may

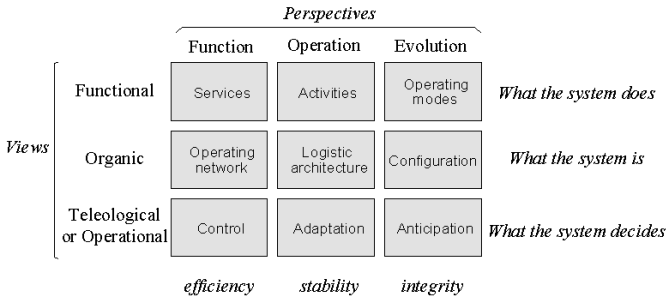

Fig. 1. The SAGACE generic matrix

be defined as follows (it is a generic definition, and must be customized for each project and each system according to the nature of the problem and the type of model to be developed) :

Service. Functional viewpoint describing the services performed by the systems, and their dependency relations.

Activities. Functional viewpoint describing the sequence of activities executed by the system to perform the expected services.

Operating modes. Functional viewpoint describing the operating modes adopted by the system and the scenarios arising from these sequences.

Operating network. Organic viewpoint describing exchanges between the organs in charge of system activities (human operators, equipment or human-machine collaboration).

Logistic architecture. Organic viewpoint describing the system architecture in the sense of the organization allowing it to ensure its activities.

Configuration. Organic viewpoint describing the overall states through which the system passes during operation, and the sequence relating these states.

Control. Operational viewpoint describing the decisional logic related to system performance.

Adaptation. Operational viewpoint describing the decisional logic of system adaptation to environmental disturbances.

Anticipation. Operational viewpoint describing the decisional logic of strategic anticipation of system evolution during operation.

This knowledge representation method has been described in detail by Penalva [1997]. The purpose of this system modeling approach is to produce a representation constituting a structured medium for information of different types from a variety of sources, a basis for collective discussion and argumentation, the concrete expression of shared knowledge of the operating situation. As the actual process equipment items in service may vary during operation, in our industrial prototype the supervisory system is so capable of recognizing the current configuration (Configuration viewpoint) and operating mode (Operating Modes viewpoint), which are used for online selection of the relevant knowledge input for the supervision and human-machine interface (HMI) modules, as well as to control their status. More details concerning this application point are provided in Section 4.

3.4. SAGACE and its Graphical Language

SAGACE provides a graphical language designed to facilitate the system representation, to allow communication among the relevant players and to materialize the shared knowledge of a subject. We imagined a language of ideographs to represent the relations among the three entities (processor, flow and ob-

server) through transactions, interactions and coupling phenomena.
The basic elements of the language express the idea of a processor carrying out transactions with its environment (horizontal axis) and submitted to the influence of the environment by interactions (vertical axis). Some of the flow characteristics are perceived by operators (ellipse).
With regard to representation, the SAGACE language uses three symbols with limited signification (the box, the arrow and the ellipse, selected for their cultural connotations) to construct patterns with increasing signification : transaction, interaction and coupling diagrams. These patterns are then defined by placing them in a viewpoint as defined above (subsection 3.3) and describing them according to the conventions adopted for each study.
The basic symbols of the SAGACE language (Figure 2) are :

- the box, a closed region delimiting an interior and an exterior, associated with the notion of processor (the site of a transformation) ;
- the arrow, denoting a transfer and associated with the notions of flow or influence ;
- the ellipse, a metaphor for the eye, associated with the notion of observer (change of level, from concrete to abstract) ; in some viewpoints, the observer may correspond to a sensor.

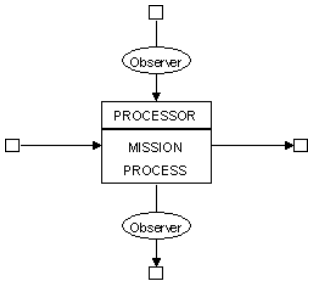

Fig. 2. Basic elements of the graphical language

These three symbols may be combined using simple rules that allow for object types.
Any action is seen as a process of transformation in time (storage), space (transport) or form (conversion) of matter, energy or information.
The implementation of the action is represented by the notion of the processor :

- an identified device that can be recognized and named ;
- characterized by a stable form, meeting specifications and thus described by properties that are either inherent or inherited from a more general form ;
- finalized, i.e. active within the scope of a mission;
- immersed in an environment on which it acts, and limited in a space-time field that it influences.

A complete system may thus be represented by a processor, in the form of a general specification indicating the system inputs/outputs, constraints and objectives considered pertinent to the study. This comprehensive overall perspective must then be projected through the model viewpoints to obtain the detailed specification.

- from a functional standpoint, the processor symbol is used to represent a *service*, an *activity* or a *behavior mode*.
- from an organic standpoint, the processor symbol represents a set of *resources* directly contributing to the action (”operative” processor), supporting the organization of the activities (”logistic” processor) or allowing a variety of behaviors (”auxiliary” processor).
- from an operational standpoint, the processor symbol represents a *context*, i.e. a representation of the system activity necessary for active intervention.

The flow symbol is used to represent relations between the system and its environment as perceived by the modeler.
The processor input and output flows along the horizontal axis express the transactions (matter, energy, information) between the processor and its environment ; the vertical axis is reserved for interactions that condition the activity. A flow is symbolized by an arrow and identified both by a name and by an (Object, [Medium]) pair where the medium is the vehicle, matrix or form of a collection of objects (Figure 3). Thus, for example, ”LOAD / M(U+FP, H+)” designates a material flow of uranium and fission products in an acidic aqueous phase.

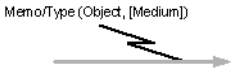

Fig. 3. Flow characteristics

The flow symbol is used to compose flow diagrams representing matter, energy and information transactions (Figure 4). In a flow diagram, ”information” refers to data undergoing processing (storage, trans-

Fig. 4. Flow and flow diagrams

mission, calculation). The flow symbol is also used in interaction diagrams also known as ”field diagrams” (Figure 5) showing :

- the *conditions* affecting the process sequence, the execution of the mission or the utilization of the processor ;
- the *events* by which the effect of the processor is known, concerning its configuration, operation or performance. Events are interpreted by the system as data that entail a response of its organization. They represent the system sensibility to the fluctuations of the fields it is constrained to.

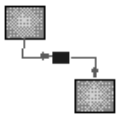

Fig. 5. Flow and interaction diagrams

Finally, a coupling diagram (Figure 6) expresses the dependencies between entities of different levels (by their complexity, their representation, their logic, their technical domain, etc.). As with any model object, the

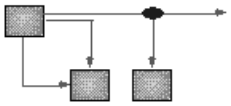

Fig. 6. Flow and coupling diagrams

flow is characterized by a list of properties. An observer (shown by an ellipse attached to an arrow) represents the ability to perceive a flow property. An ob-

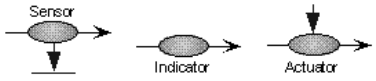

Fig. 7. Observers

server (Figure 7) may be placed on any type of flow, transaction or interaction. Its particular meaning depends on the viewpoint in which it is placed. From an organic standpoint, it represents a sensor or an actuator ; from a functional standpoint it represents a particular specification ; from an operational standpoint it is a panel indicator or represents an objective assigned to a concrete action.

Given the size of the application, it was essential not only to structure the knowledge required for all the supervision system functions into operating system viewpoints, but also to provide a unique representation method for each viewpoint. Having thus defined the formal representation framework, the methodology adopted to design and implement our industrial prototype is discussed in the next section.

4. The Demonstrator

The methodology is based on the principles and on the technical and strategic observations discussed in the preceding sections. The following paragraphs examine the models adopted for supervisory purposes based on the actual available knowledge of the facility, and show how the operating principles are incorporated into the supervision strategy through a managed combination of different models corresponding to several process control viewpoints.

As a consequence the following description does not focus on the mathematical aspects of the application (which are not necessarily the most difficult part in an industrial application) but rather on the knowledge to gain, collect and organize to achieve the supervision system.

4.1. Scope

By the size and complexity of the process facilities and their high degree of computerization, nuclear fuel reprocessing is particularly well suited for supervision applications. The scope of this application is limited to the extraction and scrubbing columns, which are situated in their industrial context including its operating modes and configurations. It is a continuous process with slow dynamics. The process solution from another facility in the plant (a nitric acid solution containing uranium, plutonium, fission products (FP) and technetium) is initially stored in a buffer tank to allow large process batches from the upstream facilities to be handled in smaller batches in the downstream feed tank.

The feed tank is automatically supplied with batches of the process solution (U+Pu+FP+Tc+HNO3, water) from the buffer tank, and in turn supplies the first extraction column, the purpose of which is to separate most of the fission products from the process stream. The FP scrubbing column eliminates any traces of

fission products (U+Pu+eFP+Tc+TBP, dodecane) entrained in the solvent from the extraction column. This nitric acid rinse allows the FP traces (eFP + HNO3, water) to be recycled in the extraction column, and supplies the Tc extraction column with a solvent feed stream from which all the fission products have been removed (U+Pu+Tc+TBP, dodecane).
The Tc scrubbing column separates the technetium from the process stream by entraining it in the aqueous phase, together with traces of uranium and plutonium. The main organic phase (U+Pu+Tc+TBP, dodecane) is sent to another facility not included in the scope of this application ; the aqueous phase is sent to an additional extraction bank.
The mixer-settler bank strips the residual uranium and plutonium from the (Tc+eU+ePu+HNO3, water) outflow from the Tc extraction column. A diluent washing unit purifies the aqueous phase containing the technetium by removing any traces of TBP remaining after the additional extraction step. The Tc raffinates from the diluent washing unit are stored in a tank.
The diluent scrubbing column eliminates any traces of TBP from the aqueous phase containing the fission products (FP+eTBP+HNO3, water) from the extraction column. The extraction is performed using a diluent (dodecane).
The heel from the diluent scrubbing column (FP+HNO3, water) is continuously transferred to a storage tank ; when the tank is full, its contents are transferred to a fission product monitoring tank.
All transfers are ensured by danaids, airlifts for continuous transfers, and by pumps, jet-pumps and siphons for batch transfers. The system response time may be set at a half-hour for hydraulic phenomena and a few hours for chemical phenomena. The facility is shown schematically in Figure 8.

The system is governed by a number of operating modes (startup, shutdown, production, etc.) and operating configurations (reversion to emergency transfer device, online adjustment, etc.).
Three types of data inputs are required for process supervision :

- 320 sensors with a 30-second acquisition time (actual recorded data, played back in accelerated time on an offline computerized system)
- 5 offline chemical analyzers
- 20 descriptors (e.g. local valve position).

This corresponds to :

- 44 supervised devices (broken down into eight classes : airlift, danaid, siphon, jet-pump, pump, online mixer, pulsed column, and mixer-settler bank) ;
- 130 anomalies ;
- 311 causes of failure ;
- approximately 400 indicators which, as noted below, constitute the basic detection and diagnosis aids considered as ”processed data” or as local models (equivalent to the residuals used by automatic control specialists, symptoms of model-based diagnosis).

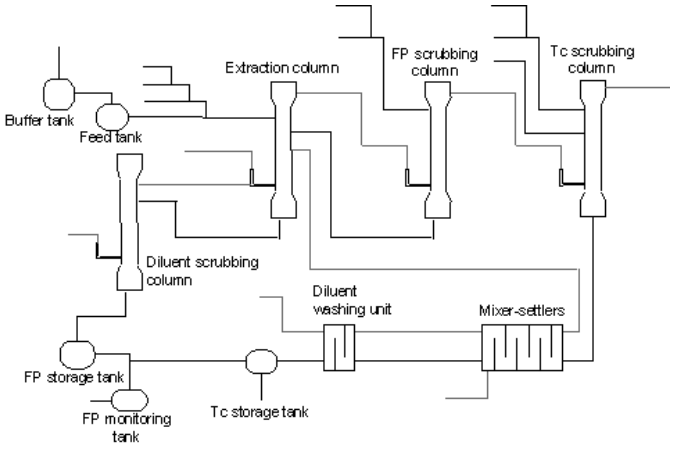

Fig. 8. Simplified system schematic

4.2. Application of the Sagace viewpoints to the plant representation

The supervision functions are implemented through the acquisition, synthesis and structuring of the data they required for application to a specific facility. In order to bring together information of various origins (process, control and monitoring, safety, etc.), we adopted the SAGACE unique representation for all the viewpoints. The knowledge partition in viewpoints simplify the way the context-dependent behavioral model is governed.
This formal framework produces the concrete expression of shared knowledge (by the operating staff and the supervision system) of the operating situation : it is essential to the intelligibility and pertinence to the operator of the results provided by the supervision system ; the results of the supervision system are part of the reasoning of the operator analyzing the situation, and are more easily substantiated and explained. In the scope of our application only three of these viewpoints are (mainly) used to capture the knowledge of the op-

erating situation necessary to our supervision objectives : operating network, operating modes and configurations.

4.2.1. Operating Modes

In the scope of the application, a *functional group is a set of devices whose operation can be controlled independently*. Automatic startup and shutdown sequences are localized at the functional group level. It is important to note that the functional breakdown does not reflect the partitioning of the facility : a device may simultaneously belong to several functional groups. Then, *an operating mode is a phase during which a functional group is in a state relevant to its objective* (and to the constraints qualifying the objective). A mode may be associated with a set of characteristics that it is desirable for the physical system to maintain. Operating mode management provides a real-time indication of the current operating phase, based exclusively on process inputs and actuators values.
This knowledge corresponds to the Functional *Operating Modes* viewpoint at the system behavior representation level, expressing a diachronic functional vision concerned with the system evolution over a large time scale during its operation.
Example. the mission of the unit X is to extract the U and Pu from a fission product solution using a solvent. One operating mode of unit X is *production*, a phase during which the extraction mission is fully ensured.
The model must provide a representation of the unit's operating modes, and the position of each mode in the unit's operating scenarios, where a scenario is an operational logic sequence of mission-pertinent operating modes. It must also support the list of information for which online acquisition is considered necessary to ensure automatic detection of the current operating mode. The operation of a functional group can thus be represented by a graph in which the nodes are the operating modes and the arcs symbolize possible transitions between modes. This SAGACE graph thus describe the possible operating scenarios.
The nodes of the general graph in Figure 9 are the nine distinguished operating modes for the application. The syntax for representing operating modes in the SAGACE method is shown in the example in Figure 10.

The operating modes are boxes ; transitions between modes are expressed as a combination of events, constraints and conditions that are logical expressions classified according to their nature :

- events allow for exiting a mode (these include objectives inherent in the mode or events originating outside the system), *examples : start danaid, column B discharged* ;
- constraints are imposed by the environment (in this case, decisions made by the operating team), *example : discharge cycle* ;
- conditions must be met before entering a new operating mode, *example : column A discharged, shutdown U(IV) and N2H4*.

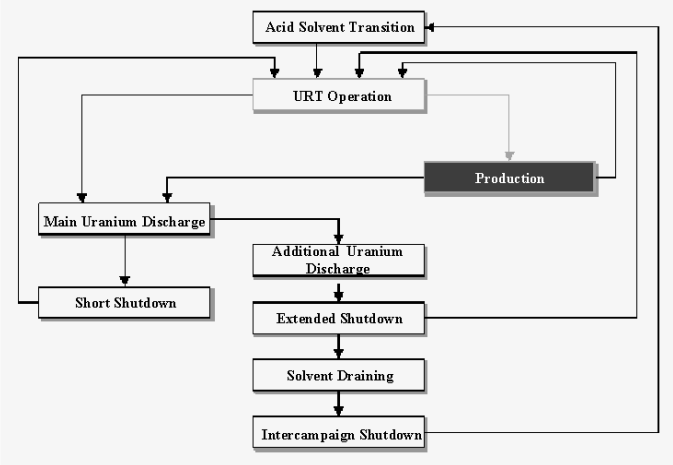

Fig. 9. General operating mode graph

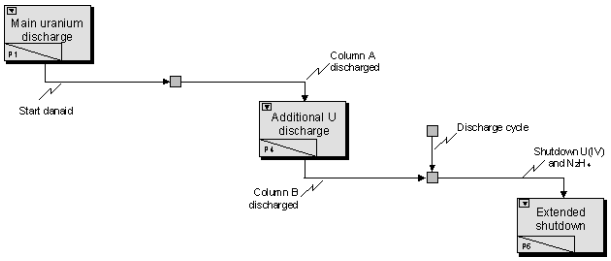

Fig. 10. Excerpt from a scenario

Theoretically a transition to detect an operating mode change is the logical product ($event * constraint * condition$). A constraint corresponds to a decision by the operating team : a logical variable that corresponds to this forcing possibility has to be declared in the model. However all the transitions do not imply a human decision : for example there is no constraint requirement in the Main Uranium Discharge mode and the Additional U Discharge mode transition, it is then reduced to the logical product $event * condition$. On the other hand events and conditions are always eval-

uated from the process data flow with a 30-second acquisition time. The simplest case corresponds to the direct acquisition of logical measurements : for example shutdown U(IV) and N2H4 condition is simply the logical product of the on/off state signals of the corresponding valves. In other case events and conditions may be associated with logical expressions as : "analogical variable measurement or a function of it (for example a density or its derivative) or a given threshold, or = a reference value".

4.2.2. Operating Configurations

An operating configuration corresponds to a selection of devices defined according to operating criteria (operating mode, equipment availability, product quality, etc.). Two types of selection are possible :

- alternatives used to obtain the basic configuration required for production (generally based on equipment availability) ;
- options implemented in addition to the basic configuration (e.g. recycling flows).

The configurations detected include devices used for redundant transfers, adjustments and implementation of optional transfer provisions.
The devices actually used may vary during operation ; recognition of the current operating configuration is therefore indispensable to the supervisory system for example. This knowledge corresponds to the Configurations viewpoint in the SAGACE matrix. The functional items capable of detecting configuration changes (e.g. on/off switching of optional or standby transfer devices) must therefore be listed.
Reconfiguration thus always involves activating and/or deactivating transfer devices controlling the use of a process line, and therefore implements purely logical models that control and adapt the supervisory behavior models. In Figure 11, air-lift A corresponds to the active transfer line from the extraction column to the PF scrubbing column whereas Air-lift B is not active and is considered as the emergency line in this configuration.

4.2.3. Organic representation

The organic model corresponds to the Operating Network viewpoint, consolidating all the information concerning the structure of the operating network, knowledge related to the instrumentation, as well as the basic knowledge for diagnosis (defect-failure graphs) and detection (anomalies). The organic model is implemented at the system structural representation level,

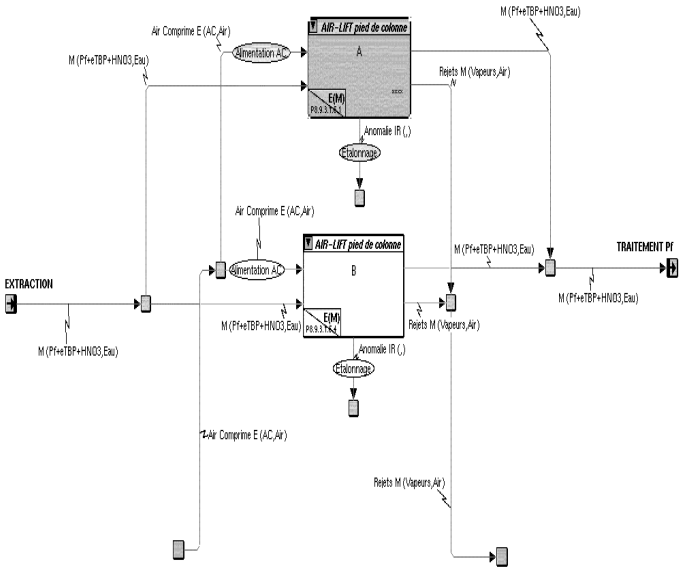

Fig. 11. Active and emergency lines

and expresses a time-independent organic vision describing the resources necessary for the production system service functions.
A hierarchical structural model of the plant is established : the highest level corresponds to the simple box entitled Spent Fuel Reprocessing Unit with its input/output flows and the lowest levels provide elementary devices. Figure 12 illustrates the organic model page of a scrubbing column complete with transfer flows. The model is automatically translated to generate the HMI views ; note that the box element symbolizing a processor corresponds here to a major equipment item, and was therefore replaced by a simple icon, as shown here for the pulsed scrubbing column. The observers designate potential device anomalies at the lowest level of the hierarchical structural breakdown (detected by the anomaly detection system). Note that thanks to this double hierarchical and flow decomposition it is as easy to surf vertically (abstraction levels) as horizontally (flows) in this computerized representation.

4.2.4. Online consequences on the behavioral model and HMI selection

The organic model, the current configuration and operating mode determine the current behavioral model. The behavioral knowledge used to diagnose failures clearly depends on the operating mode. For example, it

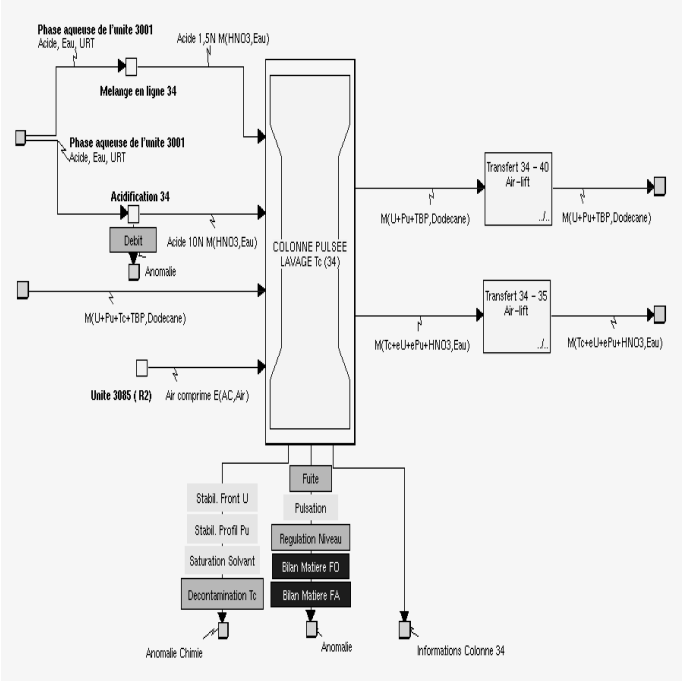

Fig. 12. Organic model of the Tc scrubbing column

may be considered useful to monitor a unit's feed system only when the unit is used for production, and the necessary calculations are possible only when a number of assumptions are verified. The operating mode recognition function for a functional group must be capable of configuring the system, i.e. of controlling its supervisory functions in a manner consistent with the current operating mode, but also covers the graphical interface display functions derived from the organic representation. The screen display must correspond to the operating context, and it may be detrimental to present information that is not pertinent to the current context : in other words, the most suitable display for a production mode is not necessarily the best adapted to a shutdown mode.

From the Configurations point of view the detection of configuration changes is just as indispensable as the detection of operating mode changes, both for reverting the prototype to standby status and for selecting the knowledge suitable for supervision of the current situation.

As a conclusion the Operating Modes and Configurations viewpoints are used to control online selection of the knowledge required to write behavior models used for diagnosis (the indicators, constructors and signs in this case, as discussed in section 4.3).

A context-dependent behavioral model is provided in the following example (Figure 13). The interest of the following extract of the knowledge base is not to present the specific language of the base : it simply relies on specific and easily readable syntax and grammar developed on the basis of C++ language and is not discussed in this paper. The idea beyond this example is rather to check that as soon as several viewpoints of the operating situations are introduced models complexity rapidly increases. The distinction between operating modes, configurations and structural knowledge becomes essential to build even apparently simple behavioral model. Indeed, from a theoretical point of view the flow balance for the extraction column can be a priori simply enunciated as : $q_{aqueous\_phase\_input}(t) + q_{organic\_phase\_input}(t) = q_{aqueous\_phase\_Output}(t) + q_{organic\_phase\_Output}(t)$. Consequently this balance that is an indicator of the extraction column class appears to be a very simple behavioral constraint. But in fact, each of this term depends on the current mode and configuration. The following extract simply illustrates the necessary calculus to process at each time to know what $q_{aqueous\_phase\_input}$ does really designate online. The same calculus method could be proposed for each inflow and outflow of the balance equation : the formal previous balance equation consequently takes multiple computational evaluation forms that are managed by the different viewpoints. The simplest expression of $q_{aqueous\_phase\_input}(t)$ is the sum of the outflow of the upstream active danaid and the feedback of the FP scrubbing column ; but to this basic value can be added additional inflows depending on adjustment configurations and recycling modes.

Class Extraction column

...
Extraction column.method
//[1] Methods for the Extraction Column class
...
//Calculated attributes
// Calculus of the aqueous input flow of the column : q aqueous_phase_input
// q aqueous_phase_input : (danaid_x_outflow or danaid_y_outflow) +
// 10% in case of recycling mode +
// online mixer_x (corresponds to the aqueous flow feedback from the FP scrubbing column) +
// online mixer_yy + online mixer_yz when acid or URT additional adjustments are needed.

method (extraction_column. q aqueous_phase_input) is
active_danaid : object (load_danaid);
inflow_active_danaid : numerical = 0.0;
recycling_inflow : numerical = 0.0;
adjustment_inflow : numerical = 0.0;

[1] // are simple comments.

```
adjustment_type : enumerated (adjustment_type);{
// adjustment type evaluation.
adjustment_type = configuration.load_adjustment.adjustment_type;
If (is_unknown (adjustment_type)) {
 return unknown_numerical;}
// Depending on the active danaid, the load inflow is evaluated.
If (is_unknown (configuration.transfert_feedtank_extractioncolumn.active_object))
{ return unknown_numerical ;}
active_danaid = cast (load_danaid, configuration.
transfert_feedtank_extractioncolumn.active_object);
inflow_active_danaid = active_danaid -> inflow_measurement ;
// Depending on the recycling mode, q % additional inflow from unit Xxx is
added to the main inflow.
If (cast (air_lift, configuration.recycling_Xxx_extractioncolumn.active_object)
== air_lift.alzzz) {
Recylcing_inflow = inflow_active_danaid * q%;}
// Depending on the additional adjustment, the additional adjustment inflow is
evaluated.
If (adjustment_type == acid adjustment) {
Adjustment_inflow = online_mixer.mixer_x.inflow_measurement;}
If (adjustment_type == urt_adjustment) {
Adjustment_inflow = online_mixer.mixer_yz.total_inflow_measurement;}
If (adjustment_type == complete_adjustment) {
Adjustment_inflow = online_mixer.mixer_yy. total_inflow_measurement +
online_mixer.mixer_yz.total_inflow_measurement;}
// q aqueous_phase_input
 return inflow_active_danaid +
online_mixer.mixer_x. total_inflow_measurement +
Recylcing_inflow +
Adjustment_inflow;}
```

Fig. 13. Context-dependent behavioral model

4.3. Application of the concept of vulnerability in the diagnosis knowledge

Supervision covers a set of functions designed to automate the process of monitoring satisfactory system operation, and to detect and account for any malfunctions at the earliest possible stages. But as explained in section 2, precocity of detection may be misunderstood by operators and has lead to distinguish plant vulnerability management and plant availability management in the application, associated to the Anomaly detection and Fault Diagnosis functionalities.

Anomaly detection. The detection of anomalies involves identifying the loss or impairment of a basic function. A degraded function does not necessarily have any consequences (immediately, at least) on the process outputs, but results in a reduction in the process control margin : anomalies are associated with the concept of vulnerability. It is a "functional" detection capable of indicating a crippled function without attempting to attribute it to a problem in the structure of the system.

Fault diagnosis. The diagnostic step attributes a malfunction to modules such as sensors, controllers, process or control units. These malfunctions affect the process outputs : the faults are associated with the concept of availability. The detection function only detects the loss or impairment of a basic process function, whereas the diagnostic function provides an explanation and determines the cause of the event. Diagnosis allows identification of the initial defect among multiple detected defects (i.e. the event capable of accounting for all the other observed defects) and determination of the physical cause.

4.3.1. Fault Diagnosis knowledge

Three main steps can be distinguished in the diagnosis knowledge base construction : failure mode and effect analysis, defect-failure graphs construction, model-based diagnosis interpretation.

Failure Mode and Effects Analysis. The diagnostic knowledge is initially based on a Failure Mode and Effects Analysis (FMEA)[2]. This analysis identifies the possible malfunctions that must be diagnosed in each process equipment item, and must be carried out with the plant operator to specify the list of failures for supervision. The study also determines the causes, effects and consequences of each failure, and assigns it a degree of gravity.

The knowledge base must include all the relevant failure modes and their possible causes for all the process equipment items covered by the application. Some items (pumps, airlifts, tanks, etc.) are found several times in a facility ; it is thus advisable to limit the diagnostic analysis to a type of generic equipment item rather than to repeat the same reasoning for multiple similar items.

This generic analysis identifies generic failure modes for process equipment. In fact, however, each device must be considered individually because of its instrumentation or its location in the facility. The list of failure modes associated with a "tank" may be determined by generic analysis, but the specific aspects of a given tank (continuous or batch feed, drain provisions, degree of instrumentation compared with the basic tank design) must be specified in the FMEA, notably the detection provisions highlighted by the generic study.

The result for an identified device may be presented as an individual data form resulting from the generic and specific studies. A typical example describing the un-

[2] Failure Mode and Effects Analysis is a qualitative system analysis method designed to identify the system component failure modes, their causes and their effects.

timely shutdown failure mode of an airlift, as summarized in an FMEA table, is shown in Figure 14.

SIGNIFICANT FAILURE MODE : Local effect :	SHUTDOWN Loss of organic phase transfer flow from CP 21 to CP 31
Failure 1 :	Loss of prime through lack of air
Failure 2 :	Loss of prime through lack of material
Failure 3 :	Inadvertent closure of AC system shutoff valve

Fig. 14. Causes and failures

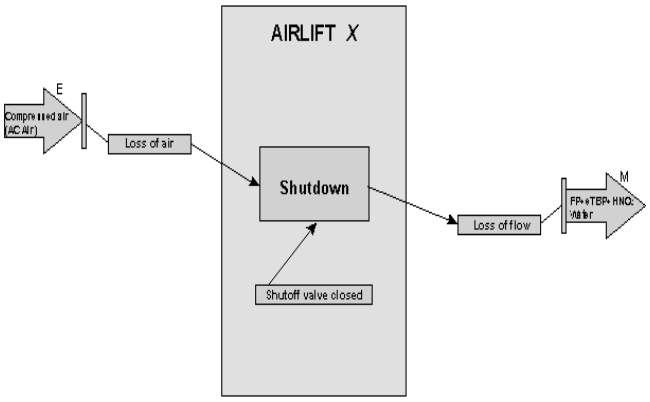

Fig. 16. Defect-failure graph of significant modes for an airlif

Defect-Failure Graphs. The FMEA tables may then be consolidated into a synthetic representation in the form of defect-failure graphs. Defect-failure graphs are structured interpretations of the FMEA. They express the causal relations between flow defects and processor failures. These graphs formalize the representation of the failure modes in a way that is directly applicable for developing diagnostic rules, and they also provide a human/machine interface for the diagnostic view : the operator can monitor the system results and reasoning on the graphs.
Partitioning the defect-failure graphs for the facility according to the gravity of the consequences they express then makes it possible to discriminate between failures with critical, significant, gradual and minor consequences for each device. The semantics for defect-failure graph is given in Figure 15 and the defect-failure graph of significant modes for an airlift X is shown in Figure 16.

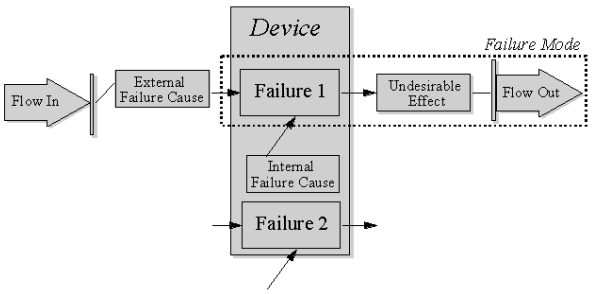

Fig. 15. Semantics of Defect-failure graph

Interpretation of the FMEA in terms of model-based diagnosis. Model-based diagnosis determines the behavior deviation between a system and its model. Deviations between the actual system behavior and the expected behavior are used to trace defective components the failure of which affects at least a portion of the process. A defect is thus an intolerable deviation between the observed and expected behavior, i.e. the "non zero residual" used by automatic control specialists. Managing this type of symptoms allows early fault detection [7].
Unfortunately, the detection means identified in the FMEA generally involve comparisons with the pre-alarm or alarm thresholds on process outputs used in conventional process control [3], implying belated detection and diagnostics : by this time, the effects of the fault condition may have propagated irreversibly, and it may be necessary to shut down all or part of the facility for a duration depending on the extent of the damage. It is therefore indispensable to reexamine each of the detection provisions in an effort to ensure safer and earlier detection ; this involves the use of local mathematical models, even if only rudimentary ones. What is necessary here is to build local behavior models to replace the detection means listed in the FMEA and thus enhance the detection performance, in an approach similar to the one proposed by [13]. For this purpose we adopted a more precise terminology and classification than the classic symptom-failure paradigm, one that prepares the formulation of the diagnostic system rule base.
A failure mode is thus conjectured when a unique logical expression, the "constructor", assigned to that mode becomes valid. The constructor may be a complex logical expression consisting of elementary logic

[3] An alarm is then a simple exceeding of a tolerance threshold by a measurement. Thresholds are generally empirically set by process engineers and do not have proper semantic : operation limits and production tolerance are sometimes equally mixed

conditions, or ”indicators”, related by conjunctive and disjunctive connectors. The indicator signifies whether or not a simple process constraint is verified, for example a discrepancy between the model and the process exceeding a specified threshold. Having defined the failure mode, the next step is to assign it a cause, which is associated with a ”context”(a logical expression equivalent to a constructor). Obviously, whenever possible the indicators are defined at the generic level (corresponding to programming classes) ; because of the specific features of some devices, however, certain methods must be explicitly defined at the device level (corresponding to programming objects).
An excerpt from the FMEA interpretation in terms of indicators, constructors and contexts of the ”airlift” device class (the concept of anomalies will be discussed later) is proposed in Figure 17. The characteristics are the specification values of a device used to construct its indicators. Certain characteristics are simple measurements, while others correspond to more elaborate assessments. This excerpt notably includes the previously mentioned instance of an unexpected shutdown.

AIRLIFT

Characteristics
SOLENOID_VALVE_CONTROL : controls the air line solenoid valve
ON_OFF_STATE : airlift on/off state
AC_FLOW : air flow supplying the airlift
ACTUATOR : air flow regulating valve opening percentage
UPSTREAM_LEVEL_SETPOINT : stabilized upstream level setpoint
UPSTREAM_LEVEL_MEASUREMENT : stabilized upstream level measurement

Q_CALC : flow rate supposed to be transferred by the airlift
Q_POT: flow rate transferred by the pot associated with the airlift
Q_STANDARD : theoretical airlift transfer flow rate

Indicators
1 - Low air flow rate
2 - Air flow rate near zero
3/4 - Shutoff valve closed
5/14 - Significant deviation between upstream level setpoint and measured value

6/15 - Upstream low-level limit exceeded
7/16 - Upstream high-level limit exceeded
8/17 - Upstream level regulating valve open
9/18 - Upstream level regulating valve fully open
11/19 - Calibration relation unverified
12/20 - Material balance for liquid-liquid separator unverified
13/21 - Material balance for liquid-liquid separator verified

Anomalies[4]
Level regulation : 5/14
AC air supply : 2 or (1 & 9/18)
Calibration : 11/19
Material balance : 12/20

Constructors and contexts
Shutdown : 6/15 or 7/16
- Loss of prime through lack of air : ((1 and 9/18) or (2 and 8/17)) and not 4
- Inadvertent closure of AC system shutoff valve : 4
Insufficient head : 11/19 and 13/21
...

Fig. 17. Indicators, anomalies and constructors

Note that the calculus of each of the indicator is context-dependent and is carried out as explained in subsection 4.2.4 (see the calculus method for the aqueous input flow of the column). Each indicator is a local model that has to be online updated by the modules performing the operating mode and configuration management.
Finally, this knowledge base could be interpreted in terms of rules. Having defined the terms, the diagnostic rule would be practically stated in the following form :

```
if <Failure Mode>[5] then assume Failure_1 with Context_1
else assume Failure_2 with Context_2
otherwise Other Failures.
```

Note that a context may well be a failure mode of an upstream device ; in fact, the context becomes the failure mode constructor for the upstream device. This allows for failure mode chaining management. Finally, the notion of ”other failures” makes it possible to list failures that cannot be associated with a context (generally because of inadequate instrumentation).

4.3.2. Anomalies and Functional Model

The malfunctions examined by the anomaly detection system do not necessarily require any intervention by the operating or maintenance crews, but they should arouse increased vigilance regarding certain portions of the facility. This function thus corresponds to the concept of vulnerability monitoring defined in section 2. Anomalies therefore do not necessarily imply the onset of failures (contrary to the diagnosis functionality), but they may constitute early symptoms of malfunctions under the failure diagnostic function. Thus, the necessary knowledge to this functionality cannot be issued from a failure modes analysis (like for diagnosis) but from a purely model based approach.
Anomalies detection is associated with a functional point of view of the plant. The mission of a process facility may be broken down into lower-level functions, each of which is implemented by a hardware de-

[4] This notion is discussed in 4.3.2

[5] If the relevant failure mode constructor is assigned

vice. Elementary functions are associated with elementary components (e.g. sensors or actuators). The system topological model is its architecture at the lowest functional breakdown level, i.e. the hardware relations. These connections express the matter, energy or information transfers between the hardware components associated with each activity, and are characterized by system description variables. The overall system behavior is determined from its topology and from the behavior of each component. Constraints among the description variables for a component constitute its behavior model ; such constraints may be expressed in various ways, from production rules to differential equations depending on the nature of the available knowledge.

As a consequence, the anomaly detection feature of our prototype detects the loss or degradation of an elementary process function that does not necessarily have any consequences on the process outputs. Any total or partial loss of an elementary function in the facility is termed an anomaly (e.g. organic material balance for a column head airlift). The anomaly (the equivalent of the diagnostic failure mode) is detected by a unique logical expression, or "sign", associated with it (equivalent to the failure mode constructor). It too is a logical expression consisting of indicators.

Consider the example in section 2, where the uranium front was assumed to descend in the extraction column. From a detection standpoint, for example, this phenomenon will be observed as a variation in the column weight differential : the elementary function corresponding to "Stabilization of the U Front" is no longer properly ensured. If the anomaly is confirmed, the diagnostic is initiated by an increase in the aqueous phase density in the lower settler and may propose a cause such as "Excessive Feed Flow".

In view of the parallels established between detection and diagnosis, anomaly and failure mode, sign and constructor, a list of all the indicators can be defined initially ; the signs (respectively constructors) can then be constructed for detection (respectively diagnosis) without rewriting indicators that may be common to both functions. It is important to reiterate that the knowledge sources for these two functions are different : interpreted FMEA for diagnosis, and functional analysis of the facility for detection. Detection, through a functional interpretation of malfunctions, is readily assimilated by the operating crew in the process control principles ; diagnosis generates structural results pertinent both to the process control staff and the maintenance staff.

4.3.3. Rule Parameter Identification

Indicators may be considered as local models involving :

- measured variables (generally subject to noise) ; the variables are assigned a variation range, and the noise is qualified by its variance ;
- calculated variables (complex functions of measured variables) ; they are assigned a noise value resulting from the noise components of the pertinent measured variables ;
- structural parameters ;
- decision parameters, i.e. thresholds beyond which operating constraints (constituting the indicators) are no longer valid (comparable to the threshold beyond which a residual is assigned a nonzero value in automatic control applications) ;
- initial values for certain variables used in differential equations.

All these quantities must be taken into account to obtain an operational base. Depending on the quality of a measurement and its implementation by the system, some variables may require preprocessing (filtering, averaging, least squares analysis, etc.). Actual data recorded during the first four weeks of operation were used during this phase to identify and validate the models.

It should be noted that statistical tests are not used for the decision parameters. Each indicator is seen as a vague statement (e.g. Air flow rate near zero, or Significant deviation between upstream level setpoint and measured value). This imprecision is modeled using fuzzy sets : the degree of pertinence of a symbolic predicate is calculated with allowance for the measured situation (What is the pertinence of the Air flow rate near zero indicator given that the measured flow rate is 27 lh-1, or of the Significant deviation indicator given that the deviation between the setpoint and the measured flow rate is 6 lh-1 ?). A degree of pertinence is thus assigned to each indicator. The signs, constructors and contexts (logical combinations of indicators) also inherit a fuzzy evaluation : the degree of pertinence of the composite expression is seen as the aggregate of the degrees of pertinence of the individual indicators composing it ; the and and or connectives in these logical combinations are then associated with fuzzy aggregation operators (t-norms and conorms) [5] [19]. This analysis is transparent to the operator.

4.4. Computerized Implementation and Validation Platform

RTWorks is used to obtain a distributed module architecture. This provides a heterogeneous environment, fault tolerance and considerable possibilities for adding or replacing system modules. A key feature of RTWorks is its ability to distribute client processes throughout the network, even in a heterogeneous environment. Processes may be run on different machines to benefit from the computing power of a network. Processes may be started and stopped dynamically while the system is running.
The Methys module, comprising the detection, diagnosis, configuration management and operating mode management functions, is an interpreter for a diagnosis-specific event-driven object language. All the modules are written in C++, Motiv and DataViews.
Actual data recordings over a period of weeks were used to validate the prototype through accelerated offline monitoring of operation ; detailed analyses were produced as forms when the operator requested a specific diagnostic report on a more significant malfunction occurring during the operating week. The spurious detection and non detection rates can be evaluated, but these case studies primarily assessed the early detection performance of the system.

5. Conclusion

In order to improve the productivity of a facility today, it appears necessary to be concerned with monitoring its availability. This has led to the development of operator aids, notably in the area of fault diagnosis. Their purpose is to ensure early detection of abnormal situations and to interpret them so that the operator may control the process more efficiently. Model-based diagnosis in Process Control Theory is founded on the construction of early mathematical fault indicators known as residuals. The methods proposed for this purpose generally represent the process by means of an extremely inflexible algebraic formalism that limits the scope of applications. Furthermore the power of these techniques is extremely poor for explanation that is a fundamental requirement for the operating staff understanding. The diagnostic procedures for such plants are generally integrated into a supervision system, and must therefore be provided with explanatory features that are essential interpretation and decision-making supports.
The introduction of early detection concept in the principles of the control staff goes through the definition of a scale of gravity in dysfunctional events. An anomaly corresponds to the impairment of a basic function of the process that does not necessarily have any consequences (immediately, at least) on the process outputs, but that simply results in a reduction in the process control margin and is associated with the concept of vulnerability. The fault diagnostic step attributes a malfunction to process modules when the process outputs are affected and is associated with the concept of availability. The sources of knowledge of both functions are clearly different : anomalies indicators are issued from a hierarchical functional breakdown of the facility whereas the diagnosis ones result from the interpretation of the failure mode and effects analysis. The distinction between anomalies detection and fault diagnosis is a simple way to introduce the chronological sequence in the scale of gravity of dysfunctional events. The notion of early detection is thus incorporated into a representation system shared by all the operators.
Targeting the supervision system on the operating team rather than the SCADA leads to other major revisions in model-based diagnosis strategies. Indeed process control theory approaches in model-based supervision are associated with diagnostics of the process itself, and not of the process control situation (the operator, facility, control triplet), which is the veritable subject of supervision. Representing different aspects of process control situation from multiple viewpoints notably allows the on line selection of the behavioral models relevant to the observed situation. It is indeed easier to allocate the wide range of knowledge necessary for a supervision system among several pertinent models rather than to build a single dynamic model that ”hypocritically” attempts to integrate all the information sources (even those for which it was not designed). Our indicators constitute the local behavioral models dedicated to anomalies and failures detection : their distributed architecture and their generic expression are adequate to the online selection and updating with regard to the different implied viewpoints.
By classifying knowledge into viewpoints, we were able to extract the operating data pertinent to the project and structure them efficiently in terms of the supervision functions required for the project : to construct a knowledge base usable not only by plant operators but also by the supervision system designers. Systemic approach SAGACE provides this formal representation framework. The purpose of this system mod-

eling approach is to produce a representation constituting a structured medium for information of different types from a variety of sources, the concrete expression of shared knowledge of the operating situation. It is essential not only to structure the knowledge required for all the supervision system functions into operating system viewpoints, but also to provide a unique representation method for each viewpoint. This representational exercise showed that configuration management and operating mode management were indispensable functions, although only the supervision of process operation had been raised during the operator's initial analysis of project requirements.

In the broader context of risk control, the operator aid cannot simply cover process malfunctions : it is intended for the human operator, and must therefore integrate a representation of human process control principles if it is to constitute an effective control room decision-making aid.

References

[1] Bainbridge, L. (1991). Les systèmes experts résoudront-ils tous les problèmes des opérateurs ? Introductory lecture : Facteurs humains de la fiabilité et de la sécurité des systèmes complexes. Ed. INRS, Vandoeuvre (France)

[2] Brunet, J., D. Jaume, M. Labarrère, A. Rault, M. Vergé (1990). Détection et diagnostic de pannes, approche par modélisation, Traité des Nouvelles technologies, série Diagnostic et Maintenance, Editions Hermès.

[3] Despres, F. (1994). Automatisation des systèmes de production du besoin à l'utilisation. Ed. Kirk.

[4] de Terssac, G. and C. Chabaud (1990). Référentiel opératif commun et fiabilité. in Les facteurs humains de la fiabilité dans les systèmes complexes, sous la direction de J. Leplat et G. de Terssac. Ed. OCTARES, Marseille (France).

[5] Dubois, D. and H. Prade (1985). A review of fuzzy set aggregation connectives. Information Sciences, 36, 85-121.

[6] Frank, P. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy, a survey and some new results. Automatica, 26 (3), 459-474.

[7] Frank, P. (1991). Fault diagnosis in dynamic system using software redundancy. Revue européenne Diagnostic et Sûreté de fonctionnement, Hermès, 1 (2), 113-143.

[8] Frank, P. (1994). Application of Fuzzy Logic to Process Supervision and Fault Diagnosis. Safeprocess'94, IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, vol. 2, 531-537, Espoo (Finland).

[9] Frank, P. (1996). Analytical and qualitative model-based fault diagnosis, a survey and some new results. European Journal of Control, 1 (2), 6-28.

[10] Gertler, J. (1997). Fault detection and isolation using parity relations. Control Eng. Practice, 5 (5), 653-661.

[11] Isermann, R. (1993). Fault diagnosis of machines via parameter estimation and knowledge processing - Tutorial paper. Automatica, 29 (4), 815-835.

[12] Isermann, R. and P. Ballé (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. Control Eng. Practice, 5 (5), 709-719.

[13] Kiupel, N. and P. Frank, (1997). A fuzzy FDI decision Making System for the Support of the Human Operator, Safeprocess'97, IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, vol. 2, 731-736, Hull (UK).

[14] Kuipers, B. (1986). Qualitative Simulation. Artificial Intelligence, 29, 289-388.

[15] Leyval, L. and S. Gentil and S. Feray-Beaumont (1994). Model based causal reasoning for process supervision. Automatica, 30 (8), 1295-1306.

[16] Millot, P. (1990). Supervision des Procédés Automatisés et Ergonomie. Ed. Hermès.

[17] Montmain, J. and S. Gentil. (1993). Interprétation qualitative pour le diagnostic en ligne. Revue Européenne Diagnostic et Sûreté de Fonctionnement, Hermès, 3 (1), 23-45.

[18] Montmain, J., L. Leyval and S. Gentil (1994). Qualitative analysis for decision making in supervision of industrial continuous processes. Mathematics and Computers in Simulation, 36, 149-163.

[19] Montmain, J. and S. Gentil (1996). Operation support for alarm filtering. Symposium on Modelling, Analysis and Simulation, CESA'96 IMACS Multicoference, Lille (France).

[20] Montmain, J. (1997). From DIAPASON research program to its industrial application in nuclear fuel reprocessing. Safeprocess'97, IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, vol. 1, 209-216, Hull (UK).

[21] MQ&D coordinated by P. Dague (1995). Qualitative Reasoning : a survey of techniques and applications, AI Communications the European journal of AI, IOS Press, 8, (3/4), 119-192.

[22] Patton, R. and J. Chen (1991). A review of parity space approaches to fault diagnosis. Safeprocess'91, IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, vol. 1, 239-255, Baden-Baden (Germany).

[23] Penalva, J-M., L. Coudouneau, L. Leyval and J. Montmain (1993). DIAPASON : a Supervision Support System. IEEE Expert Intelligent Systems and their Applications, 8 (5), 57-65.

[24] Penalva, J-M. (1997). La modélisation par les systèmes en situations complexes. Ph.D. thesis, Université de Paris XI-Orsay (France).

[25] Ragot, J., M. Darouach, D. Maquin and D. Bloch (1990). Validation de données, Traité des Nouvelles technologies, Séie Diagnostic et Maintenance, ditions Hermès.

[26] Rasmussen, J. (1993). Diagnostic reasoning in action. IEEE Transactions on Systems Man and Cybernetics, 23 (4), 981-991.

[27] Yu, C. and C. Lee. (1991). Fault Diagnosis Based on Qualitative/Quantitative Process Knowledge. AIChE Journal, 37 (4), 617-627.