PERGAMON

Engineering Applications of Artificial Intelligence 14 (2001) 441–455

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

Basic tasks for knowledge-based supervision in process control

Gerardo Acosta[a,*], Carlos Alonso González[b], Belarmino Pulido[b]

[a] *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Grupo ADQDAT, Facultad de Ingeniería, Universidad Nacional del Centro de la Provincia de Buenos Aires, Avenida del Valle 5737, B7400JWI Olavarría, Argentina*

[b] *Grupo de Sistemos Inteligentes, Departamento de Informática, Edificio Tecnologías de la Información y las Telecomunicaciones, Universidad de Valladolid, Paseo del Cementerio s/n, 47011 Valladolid, Spain*

Abstract

A new tasks taxonomy for knowledge-based global supervision (GS) of continuous industrial processes is introduced in this work. Possible required tasks are specified together with the analysis of their dimensions, which should be useful in the selection of the final capabilities of supervision. Moreover, these dimensions would help end-users and designers when comparing different systems. Several methodologies based on concepts such as generic task, generic operation or heuristic classification have been proposed to transform knowledge-based system (KBS) development in a systematic knowledge engineering activity. These approaches have been quite successful in domains such as medicine or mineral prospecting, identifying a large number of tasks that experts in the domain articulate to solve the problem. However, this was not the case in the process control area. The selection of tasks and their capabilities is the first step to be taken, even before choosing a KBS analysis and design methodology. Authors found a lack of facilities to do this selection in the aforementioned approaches when they tried to develop a global supervision tool in a beet sugar factory in Spain. Hence, this article describes an attempt to fill this gap. Moreover, it shows how this taxonomy supported the analysis and design stages of a supervision tool in the mentioned industrial application.

Keywords: Knowledge-based systems; Task taxonomy; Plant-wide supervision; Process control

1. Introduction

In applied science there exists the need for stating a methodology for the achievement of an objective. When such an objective is knowledge-based system (KBS) construction, several complimentary proposals may be found in the literature. As it is summarized in Steels (1990), there are four wide approaches to the problem of knowledge elicitation and modelling. They are Clancey's generic operations and inference structures (Clancey, 1985), the distinction between shallow and deep knowledge (Tzafestas, 1989), the problem-solving methods proposed in McDermott, (1988), and Chandrasekaran generic tasks (Chandrasekaran, 1986). These proposals were integrated in Steels' componential framework, and underlie KADS methodology (Schreiber et al., 1993). KADS I, for instance, presents the four layers experience model, identifying domain, inference, task and strategy knowledge, and introducing the concept of generic task: an abstracted experience model without the domain layer. Moreover, a generic tasks library (Tansley and Hayball, 1993) is offered as a guiding tool to KBS development, hoping that for a given problem, one may be able to find a generic task model that might act as the starting point in the building of a specific experience model.

Since KADS has been partially successful in some domains, it was initially chosen for this methodology when the problem of designing a KB supervisory system (KBSS) for the process control domain was faced. Nevertheless, the first step of system analysis, that is, initial requirement specification, posed a non-trivial question: What could a KBSS do over a continuous production plant? Several diagnosis and supervisory systems (Orsvärn, 1994; Leyval et al., 1994; Galán and González, 1994; Alarcón et al., 1994) have been proposed during the recent past, therefore the relevant literature may give some hints to answer the question. However, the first difficulty is a terminology one: there is a lack of a common language in the field of KBSS. Secondly, it is not clear which tasks must be accom-

*Corresponding author. Tel.: +54-2284-451-055/6; fax: +54-2284-450-628.

E-mail addresses: ggacosta@fio.unicen.edu.ar (G. Acosta), calonso@infor.uva.es (C.A. González), belar@infor.uva.es (B. Pulido).

0952-1976/01/$ - see front matter
PII: S0952-1976(01)00017-3

plished to supervise a continuous system: although fault detection and diagnosis seem to be necessary, they are not enough (Alonso González et al., 1998a). Thirdly, when using an accepted term, like diagnosis, or even a more specific one, like consistency based diagnosis, there are still unknowns of the problem, which are considered by the diagnostician: Does it distinguish critical from non-critical situation? Does it diagnose from a snapshot of the plant or it considers its temporal evolution?

Abbreviating this discussion, there exists the need for characterizing the problem space in the process control domain. Actually, some efforts have already been done in that direction (Isermann, 1993; Chantler et al., 1995). But both authors restrict themselves to fault detection and diagnosis. Hence, it is only a partial characterization. In this paper, global supervision (GS) from a wider point of view is considered. This is an attempt to identify, and characterize in a systematic way, an ample set of tasks that might be supported by a KBSS over a continuous dynamic system. Consequently, a first task taxonomy, partially domain independent, is proposed in this article. This has, at least, three purposes: first, to identify the subtasks that may be considered in a GS setting; second, to characterize these subtasks regarding several dimensions of concern; third, to describe and compare different systems, based on the afforded subtasks and the chosen options to support them.

An industrial application that supervises on-line a continuous plant is presented as an example (one possible approach). This description also shows how the ideas underlying the task taxonomy have influenced its architectural design.

The rest of the paper is organized as follows: in the next section a perspective on supervision is presented, then the taxonomy proposal, a description of an industrial application and the final conclusions from factory evaluation.

2. Supervision

Narrowing the analysis domain to process control,[1] supervision has two main meanings. On the one hand, it is commonly referred to as the activity carried out over a controller or a set of controllers, to assure that their objectives are fulfilled. In this case, the supervisor and the controller employ the same source of information. This is preferably termed as local supervision or supervisory control. On the other hand, supervision may be done by resorting to sources of information beyond the scope of the control law under supervision. This is best named as GS or plant-wide supervision. Thus, it will be defined as the activity in charge of checking that the trajectories of certain key process variables are close enough (as regards some pre-specified thresholds) to their desired trajectories. If this is not the case, GS must investigate further the reasons for the discrepancy and must be able to give advices and warnings to a human operator (Alonso González et al., 1998a). From this definition, it is obvious that in order to face computer-based supervision over any process, it is necessary to solve previously the control problem (the access to sensor readings and actuators). Although the access to actuators is involved, this does not mean that actuation is a constituent part of GS. The frontiers are in the output signal. Supervision is then considered as some kind of intellectual activity. This is the reason why a KBS is proposed to deal with it. Then, a supervisory system built under the scopes of this concept is not able to modify its surroundings but through a corresponding control layer.

The development of industrial software to assist the operator in GS at the control room is advantageous from several points of view, if this supervision is devoted to (aims):

- Guarantee safe operation for persons and equipment;
- assure both energetic and economical efficiencies in a minimum, over which the process must evolve;
- guarantee a pre-specified product quality;
- assure the minimum environmental aggression.

As the number of GS systems normally running up today demonstrates, they are now able to supply many facilities. Firstly, they yield more security at dangerous plants due to a non-variant attention paid to the process along time. Secondly, more uniformity in the product quality can be obtained, without the usual differences from shift to shift, and the consequent longer life of process devices. Thirdly, much less operator's stress, especially at risky processes. Finally, a better monitoring of the whole process as regards the previous planning objectives.

3. Taxonomy proposal

Eight fundamental tasks were identified as constituents of GS. The tasks and their possible interrelationships will briefly be defined. Later on, these tasks will be analyzed in more detail.

Monitoring

Given a set of variables and their desired trajectories, monitoring task identifies the subset of variables that departs from their trajectories according to some criteria.

Fault diagnosis

Fault diagnosis task has to localize faulty equipment. In addition, it may identify the causes of faults.

[1] However, this definition may be considered as general enough to be translated into other domains.

State assessment

The state of the plant is defined as a set of operation parameters that determine the factory operation, together with a set of medium- to long-term restrictions on these parameters, on some plant variables and even on plant configuration. State assessment has to establish the current state of the plant and the operation protocol adequate for that state.

Operation mode

Operation mode is the task that supervises and ensures that the plant is commanded according to the selected operation protocol.

Prognosis

This task estimates the future evolution of the plant from given initial conditions. It allows the control room operator to evaluate the effect of a set of actions over the plant.

Planning

Planning obtains sequences of actions to be done over the plant: directly, for instance changing set points through a SCADA system, or recommending actions to the operators via the human–machine interface (HMI).

Human–machine interface

HMI adaptively selects the information presented to the operators console, to avoid their cognitive overload, especially in critical situations. It must be the only interface between the operators and the whole system.

Data validation

Data validation has to filter the data received from or sent to the SCADA system or the HMI, to prevent erroneous or spurious data from proceeding to any of the previous tasks.

Fig. 1 shows these eight tasks, the data flow between the KBSS and its working environment. Communication between the plant and the KBSS is managed through the data validation task, which behaves as a data source for the remaining tasks. Hence, a double arrow could be drawn among data validation and every other task. Communication between the user and the KBSS is accomplished by the HMI. As any task may interchange data with the user, a double arrow between each task and the HMI could also be drawn. They are not drawn to clarify the figure. The data flow among the remaining six tasks has been intentionally omitted, because it depends on the scope of a particular KBSS.

For instance, a simple fault detection and identification scheme will only require the monitoring and fault diagnosis tasks, the former limited to fault detection and the latter to fault identification. On the highest degree of complexity, a system could be designed where every task cooperates on the solution of a supervisory problem, interchanging information with every other task. In such a system, state assessment task, for instance, would inform every other task which the present operation protocol. And any other task might supply information relevant for the establishment of a new plant state, including HMI as a consequence of a direct command from an authorized user.

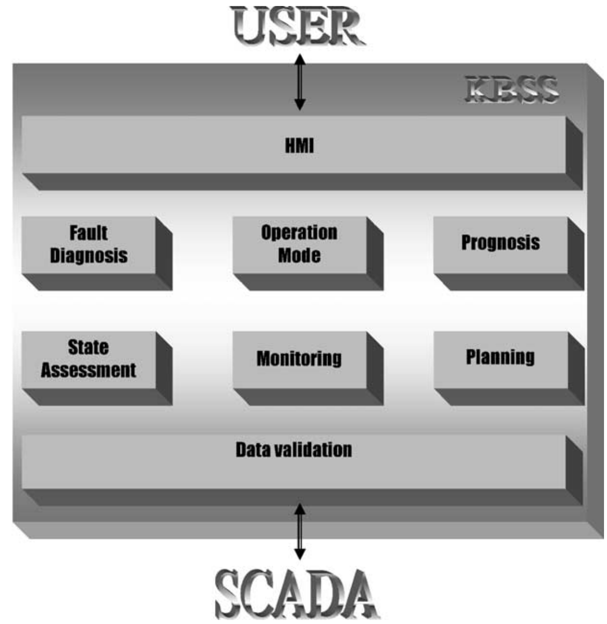

Fig. 1. The task taxonomy for a generic KBSS. Note: It is supposed that every task sends data to the HMI and that every datum from the control layer reaches the different tasks in the KBSS through the data validation task.

This task taxonomy pretends to be broad enough to describe a large set of supervisory systems. It tries to display the different possibilities available to system designers. To preserve this generality, fixed input/output description of each task is not given in this presentation. The output is implicit in the task definition. But the inputs depend on the intended scope of the system. For example, a mandatory input for fault diagnosis may be the output from monitoring; however, fault diagnosis may also consider the output from state assessment and operation mode, and even HMI and planning. In the remainder of this section a detailed characterization of these tasks will be given.

3.1. Monitoring

This task essentially compares some key variables to their desired trajectories. When one variable departs from the desired trajectory, the system must outcome one detection. As it appears from the GS definition, this is one of the first steps towards an industrial application. The simplest case of monitoring is a process measure comparison versus a fixed threshold. A more refined monitoring may include the contrast of an estimated parameter to a trajectory, mass and/or energy balances, and others. The present task, like the following ones, may be analyzed from several points of view named as dimensions in this article.

Sampling window: The detection may have taken into account the distance between measure and threshold at a single time point, in which case it is a *dot-sampling window*. If it takes into consideration a time interval (persistence), it is an *interval–sampling window*. This interval may be shifted to the past (referring to process variable historical values), as well as to the future (the monitoring "waits" for evidence until it effectively fulfills the detection), or a combination of both types.

Temporal strategy: Monitoring may be *static* if variables are compared to fixed thresholds, or *dynamic*, if they are compared to dynamic trajectories. In this last case, monitoring is capable of adapting to process' different operating points, but needs a model of it. A third possibility is *semi-static*, where variables may be compared to different sets of fixed thresholds. In this case, monitoring is able to adapt to different operation points without an explicit plant model, although it could not track the process in transients.

Information origin: This states the source of measurements. That is, *single datum* when the monitored variable is obtained from only one source, or *data fusion*, if the variable is the result of combining data coming from more than one source.

Criticism distinction: This dimension refers to whether monitoring is crucial for security or not. If the system is able to discern between variables for which a non-nominal behavior is, or is not, a potential process risk, the monitoring may be classified as *with criticism distinction*. If all monitored variables are flat for the system as regards risky situations, monitoring is said to be *without criticism distinction*.

Methodology: If possible, every monitored variable should be part of a global model, quantitative or qualitative. Then, if the set of all monitored variables may be contrasted to this model, the monitoring will be *based on first principles.* If a model is not available, the monitoring task is called *heuristic* (Fig. 2).

3.2. Fault diagnosis

Fault diagnosis will find out the causes of artifact malfunctions, that is, problems whose causes are located in process physical devices (controllers + plant). The following dimensions were characterized for this task

Sampling window: The symptoms analysis is done at a single time point, in which case it is a *dot-sampling window*, or in a time interval, that is an *interval-sampling window*.

Temporal strategy: Diagnosis may also be *static*, *semi-static* or *dynamic*. It is static if the effects are looked for in fixed thresholds. If they are compared to dynamic trajectories, with the system ability to adapt to process' different operating points, the diagnosis is said to be dynamic. An intermediate option is a semi-static temporal strategy. In general, if the system under analysis (the object to diagnose) may be considered as static, instantaneous measurements will suffice to obtain a diagnosis. This is the case of digital circuit boards, where measures are basically static due to the speed of physical phenomena involved. However, at process control domain, symptoms must be followed in their evolution along time (transients are longer and very different compared with the digital domain). Nevertheless, temporal strategic selection will depend on the troubles to diagnose: let us imagine thermal influence between two chips in the mentioned electronic board.

It is not strange that monitoring and diagnosis share dimensions, because both search for symptoms. The difference between a monitored variable and any other symptom may be stated as follows. While the former ones are signals conveying decisive information to detect

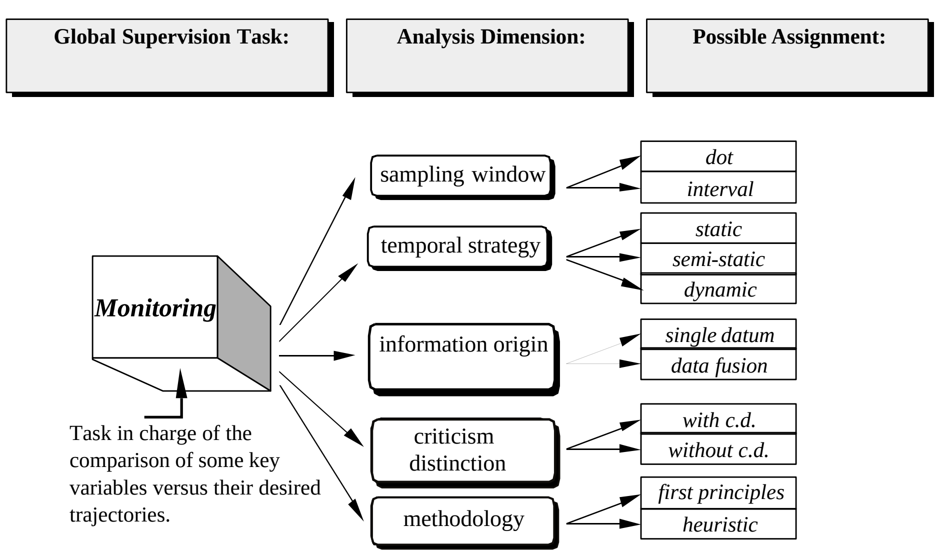

Fig. 2. Monitoring.

and then prune the causes search space, the remaining symptoms are usually employed to confirm the presence of a cause (hypotheses generation and discrimination). However, both monitored variables and symptoms are effects of one or more causes. Thus, these (additional to detection) symptoms, which are the object of analysis for the diagnosis task, contain temporal information useful for diagnosis discrimination. This is the reason why two temporal dimensions are distinguished in addition.

Temporal sequence: It intends to reflect the fact that in some cases same symptoms may suggest different causes if they appear in different order in the time axis. If the KBSS is able to evaluate this symptom arising in time, it is classified as *significant*. The opposite situation is a KBSS with *indifferent* temporal sequence.

Again, the object under analysis will determine some of these dimensions. For instance, in a static domain, with instant measurements, the temporal sequence will be indifferent. The other time-related dimension is the following.

Analysis horizon: This will be *historical*, if past values of symptomatic signals are evaluated, or *evolutionary*, if future evolution of symptoms is observed before a diagnosis is given by the KBSS. Metaphorically, if the analysis horizon is historical, the KBSS works with a "snapshot" of how the process was, while if the analysis horizon is evolutionary, the KBSS progresses in parallel with process dynamics.

Methodology: This refers to the formalisms used to carry out diagnosis. This may be *heuristic* if based on experience like most expert systems, or *based on first principles* or in domain axioms, definitions, and/or theorems, like model-based diagnosis.

Interactivity: A KBSS is *interactive* when it admits information interchange with the user during the diagnosis process or *non-interactive* in the contrary case. Sometimes, the operator may know positively that one diagnosis is not possible and informs the system about this to speed up the diagnosis process or even to change the search direction.

Criticism distinction: Again, the system may give a preferential treatment to those crucial for safety devices, in which case the KBSS is *with criticism distinction*. The opposite situation is *without criticism distinction*.

Test: This refers to the KBSS ability to interact with the process through the control layer in order to discriminate diagnostic hypotheses. In this sense, the system may be *with hypothesis verification*, if the system is able to do tests over the process to increase the evidence of a certain diagnosis. For instance, let us suppose that one candidate for explaining a problem is a jammed valve. The system may open to 100% and close to 0% successively that valve and looks for the effects over the process (or merely looks at a downstream flow meter) to decide whether this is the cause of trouble or not. If the system is not able to act in such a way, it is said to be *without hypothesis verification* (Fig. 3).

3.3. State assessment

Factories are not always running at the same operation point. Depending on input materials, market demands, state of equipment at different sections and so

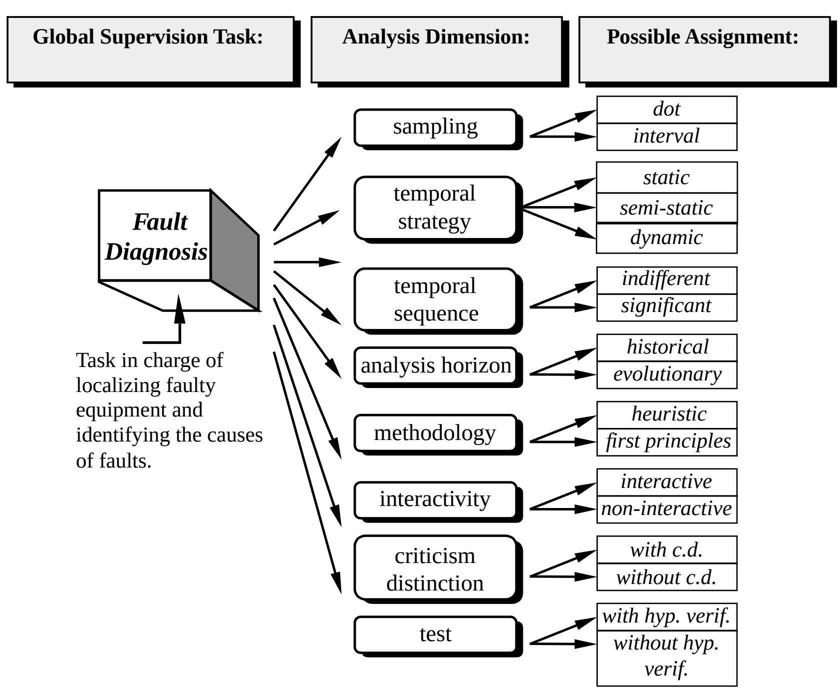

Fig. 3. Diagnosis.

on, set points of controllers have to be changed from one operation point to another. Moreover, it may be necessary to resort to different processing methods to assure the quality of the end product or to obtain other products.

Thus, an important functionality of a KBSS is to assess the state of the factory or the process, which comprises the set of parameters that determine the factory operation. For instance the desired production level, with the medium- or long-term constraints that restrict the available operation protocols. An illustrative example of restriction may be a significant reduction of the processing capacity of a section, which may require an alternative processing method, a decrease of the production level or a mixture of both. State assessment task has to determine the state of the plant and the operation protocol to operate the plant.

The process may surely seem very different at distinct states and under diverse operation protocols. Note that there will be changes in desired trajectories for important variables, normality range of inputs to functional units, and in general the functional objectives for these units.

State assessment, through the operation protocol selection, imposes different functional objectives to process units. When these are lost, the cause may be a physical artifacts malfunction, taken into account by the diagnosis task, or a deviation from the operation protocol, characterized under the operation mode task. In one of these circumstances, the system will generate a plan to restore the functional objective through the planning task (see later operation mode and planning task).

The dimensions proposed for this task are as follows.

Behavior: A *reactive* identification responds to a system need to know the present factory state, for instance because most of the monitored variables are out of the normal range. That is, some special patterns in some important signals are detected. If it is the user or other system modules supporting other tasks (i.e., planning in a scheduled identification) the one requiring the identification, the task is said to be *by request*.

Methodology: Again, as in previous tasks, it may be *heuristic or based on first principles.*

Interactivity: It may be *interactive* if it admits user's opinion during the assessment or *non-interactive* in the contrary case (Fig. 4).

3.4. Operation mode

Another common source of abnormal situations in process industry is the deviation from the selected operation protocol. This deviation may be on purpose, for instance to avoid the overflow of the next unit, or involuntary due to an operation mistake. An industrial KBSS must watch over the process in order to avoid them and, when present, detect them and give a word of advice. This task is named operation mode in this taxonomy.

Conceptually, this task aims to guarantee that the inputs to different functional units are in such a normality range that they fulfill their short-term functional objective, that is usually implicit in the operation protocol. Because the operation mode shares features with diagnosis, it also shares some dimensions. They are *sampling window*, *temporal strategy*, *temporal sequence*, *analysis horizon*, *interactivity*, *criticism distinction*, and *test*. These definitions will not be repeated again. Diagnosis and operation mode do not share methodology because for the system perceiving an operation mode error, it must have available information about plant topology (which inputs to which devices), and this has nothing to do with an heuristic model. An intrinsic dimension to this task is the following.

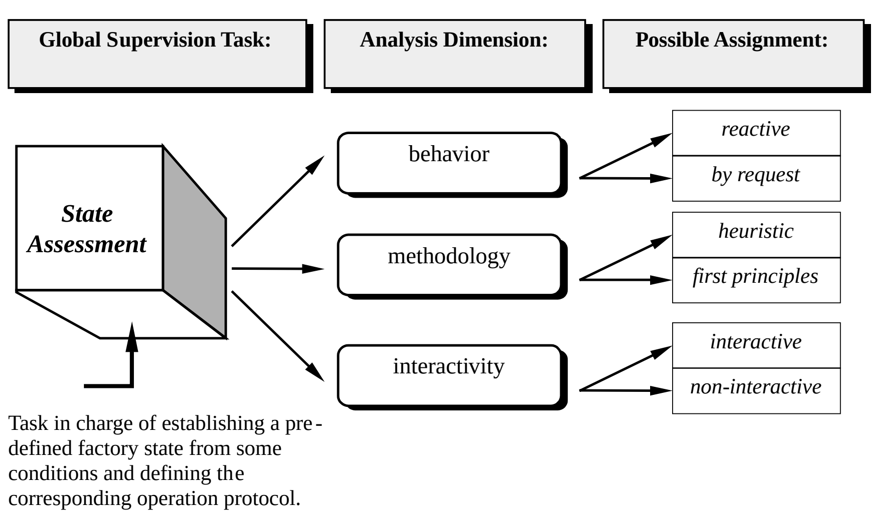

Fig. 4. Operation mode.

Topological configuration: If the system does not follow any plant topological change, the task is a *fixed topology* one. Conversely, if the KBSS auto-configures internally in real time as the plant topology changes, it is said to be of *variable configuration*. An everyday example is the case when a valve is not working properly and an operator opens the by-pass. In this case, a variable configuration KBSS must change its internal process model (Fig. 5).

3.5. Prognosis

Another task performs the consequence evaluation of some actions done over the process. It is called prognosis because an essential part of it is just to predict (in some sense) what will happen in the future, related to the evaluation time. It may be considered as the diagnosis dual task. Effectively, while diagnosis tries to find out the causes from some working hypothesis and observed symptoms, prognosis finds out the effects in the future given present causes and some initial conditions. Obviously, as always when one is claiming to predict, one needs some kind of model of the process at hand.

This task shares some dimensions with diagnosis: *methodology* (related to which kind of model the system is using), and *interactivity*. It also shares the *behavior* dimension with state assessment, which deserves some words. Prognosis may be *reactive* if the forecast or consequences valuation is done when some special patterns in some important signals are detected. This feature is vital for safety: it is known from post hoc analysis that in most great accidents in factories there were forerunner tokens which were either not seen or misinterpreted by operators. Of course, this "what-if" task may also be done *by request* (Fig. 6).

3.6. Planning

Frequently, in control room, it is useful that the operator has an explanation of what has happened or what is happening with the process. In addition, it is advantageous to assist him, or her, in order to achieve an objective in the future. Hence, planning generates operation sequences for objectives achievement or it assists the users to recover the plant from incidents. Planning shares some dimensions with previously defined tasks, such as methodology, behavior, temporal strategy and criticism distinction.

Methodology: It is *based on first principles* when the system uses some kind of model to actually build a plan. It is *heuristic* if the planner retrieves a predefined plan; planning literature usually names as reactive this kind of planning, a term that is reserved here for a type of behavior. As previously stated, a *reactive behavior* task starts up activity in reaction to some inputs. However, planning behavior may be *by request*. Let us suppose that fault diagnosis concludes that there is an out-of-

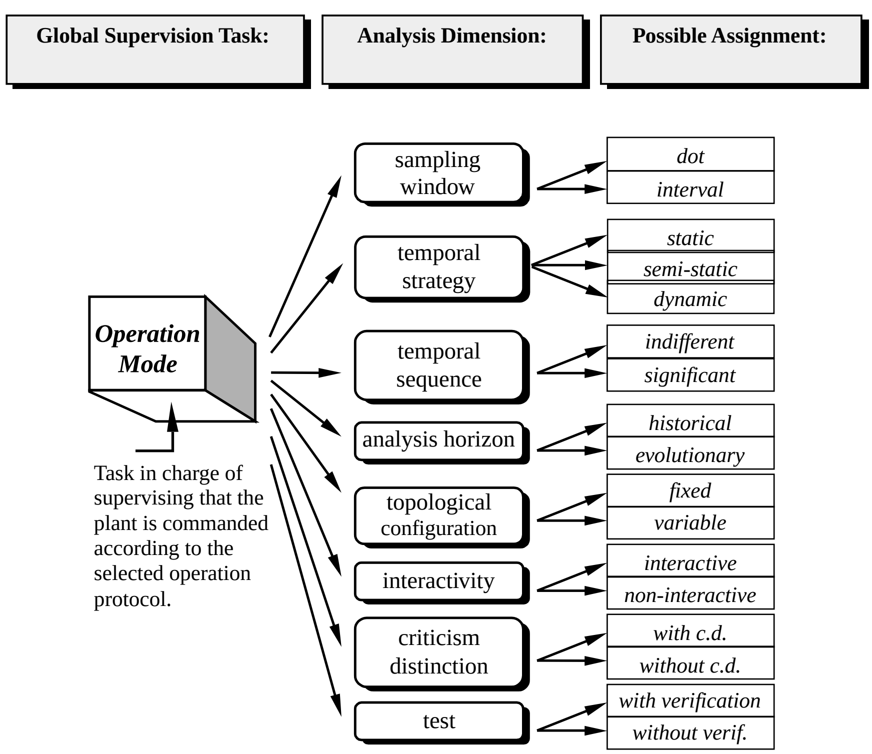

Fig. 5. State assessment.

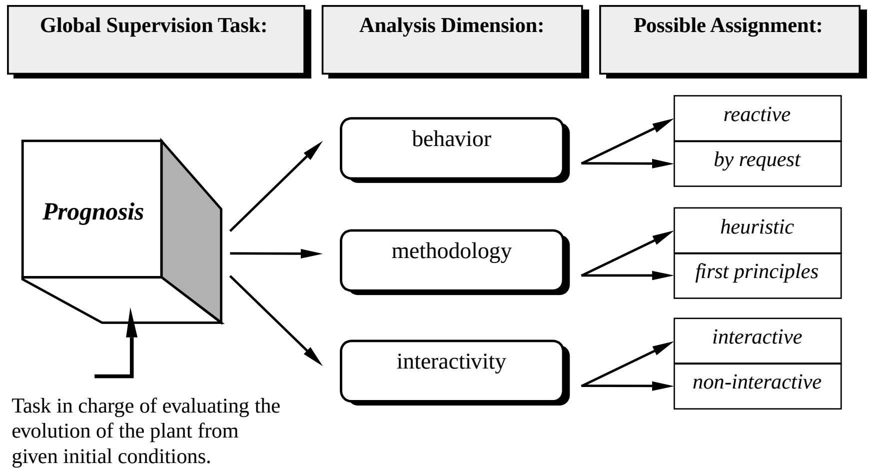

Fig. 6. Prognosis.

order device. Then, it may require according to the planning task a policy to repair it (reparative maintenance).

Temporal strategy: It may also be *static*, *semi-static* or *dynamic*. It is *static* if the planner always considers the plant at the same stationary operation point. A *semi-static* planner will adapt to different stationary points, for instance considering the output of state assessment. A dynamic planner may build a plant from any operation point, which usually will require a methodology *based on first principles*.

Planning may also distinguish the urgency of response among different situations. It is not the same to generate sequences of operations to replace a filter in preventive maintenance rather than to outcome a plan to recover the factory from an incident. This is reflected in the *criticism distinction*, introduced earlier. Other dimensions are presented in the sequel.

Purpose: This dimension is a faithful reflection of the very KBSS objectives. Then, purpose may be optimization, flux coordination, protocol change, maintenance, and repairing. The *optimization* purpose of planning is understood to be the trajectory generation in order to push the process over an optimum as regards different evaluation parameters, such as product quality, energy consumption and/or environment incidence. Another reason for planning is present when two or more sections of a process with different and independent dynamics share the same flux, and there is a physical device (i.e., a tank, a sink,…) where they converge. The control layer in this situation needs to be coordinated in order to keep sections functionality, even without the need to be optimizing versus a merit figure. This purpose of planning is referred to here as *flux coordination*. Planning may also be useful to change from a stationary working point to another. This purpose is called *protocol change*. The remaining planning purposes are related to incident management, to avoid them or to recover from them. A decisive factor for a sustained security is the preventive maintenance. In fact, in a factory there are equipment suffering from cyclic tear and wear (i.e., filters, rotating machines, heat exchangers,…), with more or less criticality as regards security, for which a preventive or programmed maintenance is highly convenient. Then, the task proposing to remove devices is called as with *maintenance* purpose. This device extraction may also respond to its malfunction, identified by the diagnosis task, in which case the planning objective (a short-term one) is to replace it, proposing the necessary steps to achieve this goal. This is clearly a *repairing* purpose.

Autonomy: The KBSS is *closed loop* as regards this dimension if it is able to connect its outputs to the control layer. On the contrary, it is *open loop* if system's output is not used to command process controllers or actuators (control layer elements). In fact, planning is the only task of GS with the ability of an interface between the inferences and the control layer, through data validation, in order to produce a change in the process (Fig. 7).

3.7. Human–machine interface

It has been stressed in this article that KBSS should be conceived essentially as operator's assistant. In this way, the task carrying out the interface between the system activity and the user must be quoted and carefully analyzed during the KBSS development to avoid cognitive overload. In this sense, HMI must take decisions based on the knowledge involved in the situation. Thus, this particular task generates outputs and is able to receive inputs such that those and these are intelligible for human beings. It is obvious that interactivity and by-request behavior, as presented in previous paragraphs, are more necessary conditions than an analysis dimension.

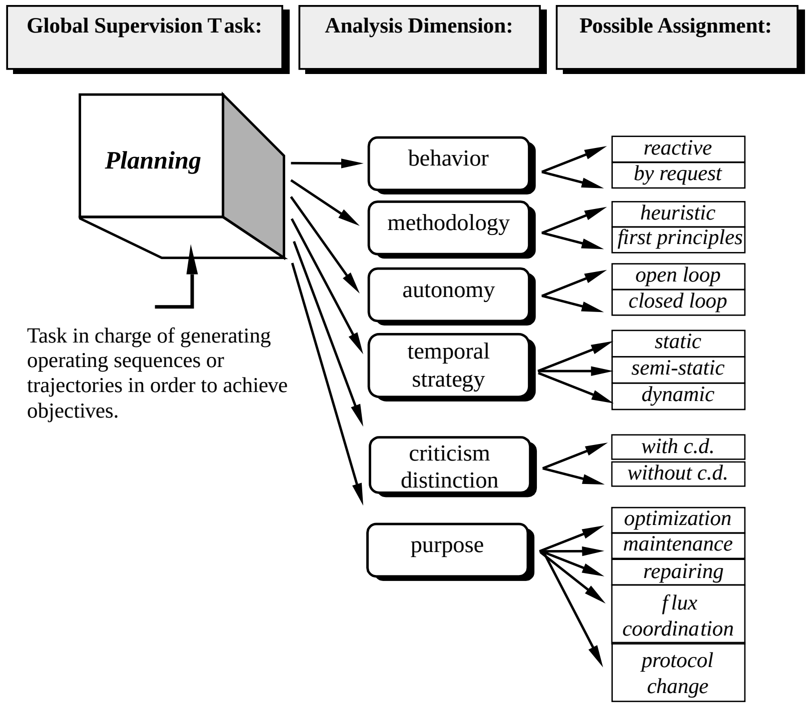

Fig. 7. Planning.

There are other characterizing dimensions proposed as follows:

Permanence: It is related evidently to time, and with the interface physical support. Such support may be volatile or persistent, and then the permanence will be *ephemeral* or *perennial*, respectively. An ephemeral HMI task is for instance a screen display of a plant mimic (in general, lasting a short time). If the physical support is for instance the system's hard disk or a printer paper, the HMI task is said to be perennial.

User attention level: This task must filter in a varied degree the information given to the user, as already mentioned. Some systems offer different levels of communication with the user when required. Others have a unique degree of information filtering to avoid cognitive overflow. Those are said to be with a *multiple user attention level* and these with a *single user attention level*.

Criticism distinction: A dual to prior situation is present in the information exchange between HMI and other tasks developed within the KBSS. In some situations (i.e., critical situations like a broken device or the necessity of a sudden stop of the process), they can ask HMI to directly show the user its outcomes. That is, if the task giving rise to the activity is with criticism distinction, it may settle this in its outcome. Then, if HMI is also *with criticism distinction*, the communication to the user will be different than if it is without this feature. In the first case messages may for instance directly put in the KBSS screen and in the last one, it may follow another route. Note as well that HMI may be *with criticism distinction* although the other tasks were *without* it. In this case, criticism distinction only affects the set of current messages that are actually considered to be sent to the user.

Profile distinction: Not every KBSS user may understand the same kind of messages or may need the same information. The system might not show the same behavior when interacting with a plant operator or with the factory director. This is managed with user profiles. Then, as regards its HMI a KBSS can be *with profile distinction* or *without* it (Fig. 8).

3.8. Data validation

It is very well known that raw data must never be used in safe inferences. Then, a previous preparation of them must be done. Another important fact to take into consideration is the great amount of field data in an average industrial process. They are more tractable if they are well structured around knowledge objects. Thus, this is another task that may be managed by the knowledge-based approach. Data validation is then defined as the task of preparing process data for inferences. The distinguished dimension analysis included the following.

Update rate: A datum may be requested by the KBSS with a pre-defined sample rate, in which case this task is said to be with a *synchronous update rate*. This situation may be possible if there are few variables to employ in

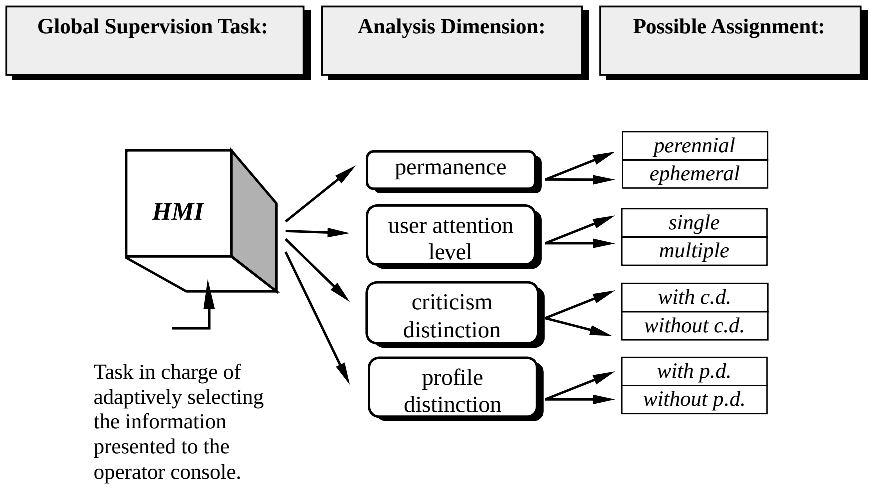

Fig. 8. Human–machine interface.

the inferences. However, in an average industrial application this is impossible. It is worth going for the datum whenever it is needed for an inference. Once updated, it may have a validity interval. This is named as *asynchronous update rate*.

Information origin: As in monitoring, variables given for inferences to the KBSS may be the result of combining or not combining different measures from field sensors. Then, the possible assignments for this analysis dimension will be *single datum* and *data fusion*.

Methodology: Data may be validated based, preferably on some model, structured like a data reconciliation from energy and mass balance, or unstructured like a neural network based model. Even a simple range and derivative checking will be a simple but useful constraint model. They are instances of a methodology based on *first principles*. *Heuristics* are also often used in the determination of an erroneous measure (i.e., pH sensors) (Fig. 9).

4. Brief description of an application

The ideas presented in the previous paragraphs were employed in the design and development of a KBSS for a beet-sugar factory in Benavente, Zamora, Spain (Prada Moraga et al., 2000). The process is an average one, with about 400 control loops. These loops are commanded from a control room via a distributed control system or DCS (Siemens' Teleperm®). The KBSS runs on a Sparc® WS under Unix®, communicating with the process through a real-time data base (RTDB). This RTDB stores data directly coming from field sensors and resides in another computer. By so doing, any stop in the KBSS for maintenance or for the upgrade of any module, does not affect the periodical data collection. The laboratory personnel may also have access to the RTDB to introduce off-line variables (i.e., from chemical analysis) or to read on-line variables (i.e., pH value in a tank from a sensor reading). Both the KBSS and the RTDB host machines are connected to the proprietary Teleperm bus.

The KBSS was developed using a specific shell, Gensym's G2®, which is very well prepared to face real-time problems, and for fast prototyping. The successive prototypes were first tested in simulation at the University Lab, and then set at Benavente's factory. This final adjustment varied from simple thresholds tuning to a complete editing of an alarm or an HMI change for a better operator acceptance.

The industrial process consists of four main sections: diffusion, depuration, evaporation and crystallization. Raw pieces of beet go into the diffusion section to obtain a juice rich in saccharose ($\geqslant$15%). This is achieved in the diffuser, a cylinder of about 25 m length by 8 m diameter, which is rotating at about 28 rpm, receiving the beet at one extreme and water at about 75°C. The impurities (non-sugars) obtained at the diffuser are eliminated in the purification, in order to obtain a juice of great purity ($\geqslant$95%). The next stage, evaporation, increases the juice concentration by extracting water by boiling, and producing syrup ($\geqslant$65% saccharose). The evaporation section consists of 9 evaporators in series, of the type Fallstrom and Robert. Finally, concentration is further increased in bath units until sugar crystal can grow in an over-saturated environment.

The KBSS operates mainly over diffusion, depuration and evaporation and it supports, to a different extent, all the proposed tasks. Initially, it was designed to supervise the factory under nominal conditions (Pulido Junquera et al., 1998), and state assessment task was not considered. Afterwards, the system scope was extended via state assessment (Alonso González and Rodríguez

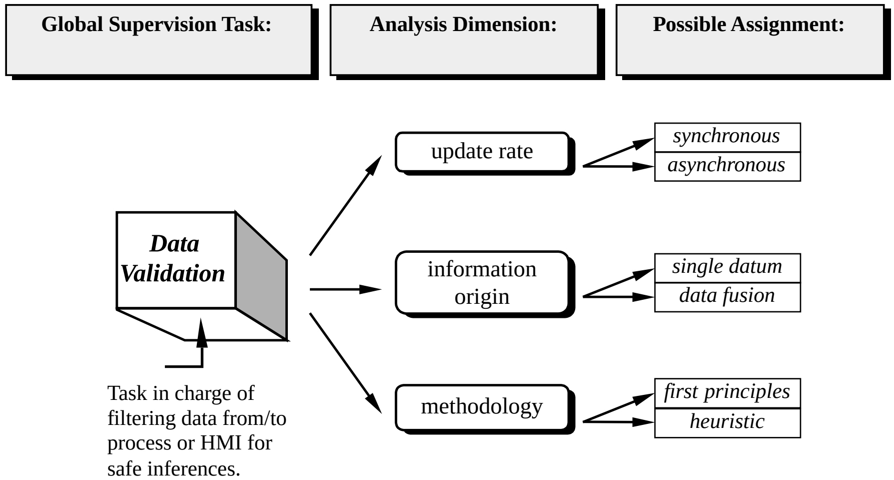

Fig. 9. Data validation.

Diez, 1999) to allow the KBSS to supervise the factory under different working conditions. Fig. 10 shows the information flow among the basic tasks. A strong hierarchy relates the main tasks: the operation protocol established by state assessment modifies the behavior of monitoring, operation mode, fault diagnosis and planning, but none of these tasks influence the former. Moreover, each particular detection established by monitoring, was considered first by operation mode. Fault diagnosis was invoked only if the related units were operated according to the selected protocol.

To cope with the complexity due to the large number of components found in the aforementioned sections, monitoring, operation mode and fault diagnosis relied on a hierarchical decomposition of the whole plant. This policy transforms a complex problem into multiple simpler (although interrelated) subproblems. The plant was conceptually divided into small and partially overlapping areas, associated to monitored variables. These variables (Alonso González et al., 1998a) were signals that convey information related to the plant behavior. They were selected by the process engineering team, taking into account the amount of information they convey, to monitor as few variables as possible and the nature of the problems that they can detect (to quickly detect important faults). Monitoring and operation mode worked locally on each region, while fault diagnosis took into account the interaction among adjacent areas.

In the sequel, the tasks will be described as they were implemented for the present application, characterizing them in terms of the previous taxonomy (see Fig. 10).

Monitoring

This task checked the trajectory of monitored variables against fixed thresholds. A hysteresis mechanism involving three thresholds, temporal persistence and possibly additional knowledge yields a robust detection scheme. The task is supported by a set of rules,

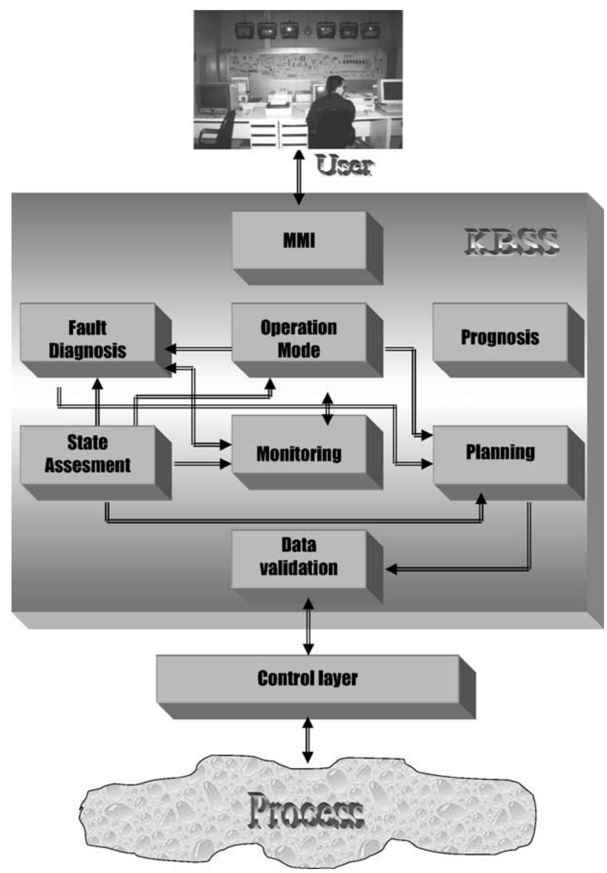

Fig. 10. The tasks with the information flow of the implemented KBSS. Note: As in Fig. 1, it is supposed that every datum from the control layer reaches the different tasks in the KBSS through the data validation task, and that every datum from and to the user goes through the HMI. This information flow is not depicted in Fig. 10 for a better comprehension of it.

periodically and independently invoked for each monitored variable. When a detection is established, the state of the related monitored variable is changed to critical, which is the output of this task. Regarding the proposed taxonomy, it could be described as:

- Sampling window: interval, because it compares time series against thresholds.
- Temporal strategy: semi-static, because the thresholds are fixed, but may be different under distinct operation protocols.
- Information origin: single datum and data fusion, depending on the nature of the monitored variable.
- Criticism distinction: without criticism distinction, because every potential problem was monitored with the same priority.
- Methodology: heuristic, relying on expert knowledge.

Operation mode

The desired settings for a particular operation protocol were stored in look-up tables (Pulido Junquera et al., 1998), which allows instantaneous checking of valid set points, and control schemata. Controller tunings were also supervized. Operation mode was invoked whenever a monitored variable became critical, launching a set of procedures that sequentially checked the pertinent actual settings against the look-up table. The specific dimensions are:

- Sampling window: dot and interval; while a set point can be instantaneously checked, a controller tuning requires examining a time series.
- Temporal strategy: semi-static, since all the settings were fixed in advanced in every operation protocol.
- Temporal sequence: indifferent; actually, on the KBSS implemented and related to operation mode, this dimension could only be applied to controller tunings.
- Analysis horizon: historical, considering only past data.
- Criticism distinction: without criticism distinction.
- Interactivity: non-interactive, because the user does not intervene on operation mode.
- Test: without hypothesis verification, since planning does not propose further tests to confirm a mode operation problem.
- Topological configuration: fixed; a limited amount of topological changes were considered as structural failures and managed by fault diagnosis task.

Fault diagnosis

Fault diagnosis was invoked for monitored variables in critical state only if mode operation did not identify a deviation from the operation protocol that could affect the variable. Fault diagnosis associated a set of rules to each monitored variable, which tries to localize and identify the fault. Rule invocation was managed by a diagnostic protocol (Pulido Junquera et al., 1998) that considered two phases: instantaneous, where a diagnosis is obtained based on historical symptoms or former diagnosis, and waiting, where the system waits for expected symptoms to refine the diagnosis. Fault diagnosis made use of single fault hypothesis per monitored variable. Hence, the diagnosis process ended when a diagnosis was found. If none of the possible candidates could be asserted, the output of fault diagnosis task was an unknown diagnosis. In terms of the task taxonomy, it could be described as:

- Sampling window: interval, because symptoms could be searched in a wide interval around the detection time.
- Temporal strategy: semi-static, because symptoms are established looking for values of plant variables in fixed regions, predefined for every operation protocol.
- Temporal sequence: significant, because the temporal order of symptoms was meaningful.
- Analysis horizon: evolutionary, because the diagnosis protocol could stay watching the plant progress.
- Criticism distinction: without criticism distinction.
- Methodology: heuristic, based mainly on expert knowledge, as in the previous task.
- Interactivity: non-interactive.
- Test: without hypothesis verification.

State assessment

On this application, state assessment was conceived as a cooperative task among end users and the KBSS. To support this decision-making, a graphical rule language (Alonso González and Rodríguez Diez, 1999) was developed. This language allows the operators to graphically define the conditions to change the current state or the current protocol; it also generates graphical explanations. State assessment considered three basic states: nominal, semi-low and low, with several restrictions on the nominal state. Each pair basic state and restriction was associated to a unique operation protocol. A protocol determines basically the process technique to be used, which may involve topological changes, set points for the main controlled variables and security limits. Each protocol selectively activated and deactivated variables to be monitored, and could change thresholds for monitoring and fault diagnosis and picked different look-up tables for mode operation. Further on, the main steps to change from one protocol to another were summarized by planning. The three dimensions for this task are:

- Behavior: reactive and by request; reactive because it was invoked periodically and by events, by requests because it could always be invoked by the users.
- Methodology: heuristic, based on expert knowledge.
- Interactivity: interactive, since the final assessment is left to the users. Actually, authorized users may impose a new state and protocol independently of the system advice.

Prognosis

Prognosis was conceived as a "What if?" facility for end users de Prada et al. (2001). It is based on stationary model of the plant, obtained from mass and energy balances, and dynamically adjusted to different stationary conditions. It allows estimating the long-term effect of a change on the main set points of the plant: rate of fresh water to diffusion, brix at the output of the evaporation section, and similar decision variables. The dimensions for this task are:

- Behavior: by request, because it is always invoked by the operators.
- Interactivity: non-interactive.
- Methodology: first principles, inasmuch as it is based on fundamental equations.

Planning

Planning was the least developed task and it was mainly limited to recall the operators main steps to change from one operation protocol to another and to substitute devices for maintenance and repairing. The dimensions for this task are:

- Behavior: by request, under some other module demand.
- Methodology: heuristic, with a direct association between the request and the selected plan.
- Autonomy: open loop, since its main outputs were recipes for plant operators.
- Temporal strategy: static, because the task did not take into account the current plant state, excepting recommendations related to protocol changes.
- Criticism distinction: without criticism distinction.
- Purpose: protocol change, maintenance and repairing.

Human–machine interface

In this application, a special effort was made to selectively show the operators the most significant information. The user attention level was dynamically changed, reducing it in crucial situations to restrict to a minimum the generated messages, considering their criticality. HMI was also responsible for presenting in a unified mode the related output of different tasks, for instance a detection from monitoring and its diagnosis from fault diagnosis. In terms of the task taxonomy it could be described as:

- Permanence: ephemeral and perennial, because although the larger amount of information was output to the screen, a shift report with all incidents was printed.
- Behavior: reactive and by request.
- User attention level: multiple, set by the users, with the possibility of being changed by the KBSS.
- Criticism distinction: with criticism distinction, attending to the predefined criticality of each problem.
- Profile distinction: with profile distinction, selected by authorized users.

Data validation

All the data from or to the plant, and numerical data from the users through the HMI, were tested by the data validation task. It was supported by a constraint check that limited the value and first derivative of physical variables. Enumerated variables were checked against its type. Non-valid values were discarded and requested again. The mean values of the main flows were tested by stationary mass/balance equations. The dimensions for this task are:

- Update rate: asynchronous, whenever an inference needs fresh data, and synchronous, in case a fixed sample rate is needed (i.e., when dealing with historical data).
- Information origin: single datum, specific process data.
- Methodology: first principles, like mass conservation and physical constraints of magnitudes.

5. Tests at factory

Quantitative measurement of the overall performance of a KBSS is a very difficult task. However, some of the supported tasks described earlier were systematically tested via questionnaires that the end users must complete on line. Particularly for monitoring, fault detection and mode operation, different and promising results were obtained at Benavente Factory, which are reported elsewhere (Alonso González et al., 1998a, b; Pulido Junquera et al., 1998), and partially transcribed in the sequel.

From a total of 32 fault detections, 27 (84.4%) were correct and 5 were incorrect (15.6%). Of the correct 27 detections, 22 yielded correct diagnosis, that is, right trouble identification (81.5%), 3 incorrect diagnoses (11.1%) and 2 unknown causes (7.4%). As regards these last, the unknown cause may be also considered a useful KBSS outcome for the operator, because although not precisely, the system gives information about some problem in a factory section that effectively has it.

Since the end of the project for this KBSS development, the University team is no longer in charge of the system maintenance. Then, results obtained thereafter from the everyday running of the KBSS, falls into private property of the sugar company.

6. Conclusions

Surely, this taxonomy is not exhaustive. Nevertheless, it should be considered as a helpful contribution in the first stages of development of knowledge-based supervisory systems. It must also be quoted that the organization around the introduced set of tasks facilitates a natural system construction in modules. These modules can be built independently, sharing a common memory like in a blackboard deployment or in agents architecture, to interchange data and/or knowledge. Even further, the implementation presented here is just one among the possibilities that might be done with the task taxonomy given. Of course, different strategies selected from the taxonomy for the task implementation through these modules will give rise to very different system behavior.

This work may generate discrepancies whether a task set should be classified in one way or another. Furthermore, the work of the proposal of "such methodology for such task" is still pending, but it should be elaborated from known and accepted libraries of problem-solving methods like those that are proposed in KADS. Even the problem-solving method for many of these tasks is yet an open research field. Moreover, it may be interesting to share a common language to compare existing KBSS and to define from scratch a KBSS for a process industry. This also facilitates the documentation in the development phase of such a system.

An other fact to consider is that a KBSS was designed, developed and is now at normal running in an industrial environment since 1994, as another tool in the factory control room. As Mamdani pointed out in Mamdani (1994), the results of an application may not induce to legitimize any theory (real situation conditions are seldom never the theory axioms and then scientific method cannot be applied straightforward). However, it is very important for AI research to achieve technological applications born at the light of AI theories, especially if they are tested in industrial fields as in the present approach.

Acknowledgements

The authors are grateful to the indoor personnel of Sociedad General Azucarera de España, S.A. (now Azucarera Ebro Agrícola, S.A. (AEA S.A.),) for their cooperation and active participation. They are particularly grateful to G. D'Aubarède, M. Bollain, R. Vargas, J. Barahona, L. Lorente and A. Bastos. This article was written under the support of "Proyecto CYTED VII.5: Técnicas de Inteligencia Artificial en Supervisión, Diagnosis, y Control de Procesos Continuos" and CICYT projects TAP98-0820 and TAP99-0344.

References

Alarcón, J., Alamán, X., Gómez, P., Romero, S., Aguilar, J.A., Campos, M., Serrahima, P., Domínguez, J.M., Rodríguez, P., DePablo, E., 1994. An integration methodology and architecture for intelligent systems in process control: the HINT project. Proceedings of the IFAC Artificial Intelligence in Real-time Control Symposium, AIRTC'94, Valencia, Spain, pp. 549–554.

Alonso González, C., Acosta, G., Mira, J., de Prada, C., 1998a. Knowledge based process control supervision and diagnosis: the aerolid approach. Expert Systems with Applications 14, 371–383.

Alonso González, C., Pulido Junquera, B., Acosta, G., 1998b. On Line Industrial Diagnosis: an Attempt to Apply Artificial Intelligence Techniques to Process Control, Springer Lecture Notes in Artificial Intelligence, Vol. 1415. Springer, Berlin, pp. 804–813.

Alonso González, C., Rodríguez Diez, J.J., 1999. In: Mohammadian, M. (Ed.), A Graphical Rule Language for Continuous Dynamic Systems. Concurrent Systems Engineering Series, vol. 55. Computational Intelligence for Modelling, Control and Automation. Evolutionary Computation & Fuzzy Logic for Intelligent Control, Knowledge Acquisition & Information Retrieval. IOS Press, pp. 482–487.

Chandrasekaran, B., 1986. Generic tasks in knowledge-based reasoning: high-level building blocs for expert system design. IEEE Expert, Fall, pp. 23–30.

Chantler, M.J., Shen, Q., Leitch, R.R., Coghill, G.M., 1995. On choosing candidate generation and prediction techniques. Proceedings of the Sixth International Workshop on Principles of Diagnosis. Gosslar, Germany, pp. 49–56.

Clancey, W.J., 1985. Heuristic classification. Artificial Intelligence 27, 289–350.

Galán, J.J., González, A., 1994. The Development of an Artificial Intelligence Real-Time Toolkit: REAKT. Proceedings of the IFAC Artificial Intelligence in Real-time Control Symp. AIRTC'94, Valencia, Spain, pp. 527–533.

Isermann, R., 1993. Fault diagnosis of machines via parameter estimation and knowledge processing. Automatica 29, 815–836.

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

Mamdani, E.H., 1994. A Misconception of Theory and Application. IEEE Expert, August, pp. 27–28.

McDermott, J., 1988. A taxonomy of problem-solving methods. In: Marcus, S. (Ed.), Automating Knowledge Acquisition for Expert Systems. Kluwer Pub, Dordrecht, pp. 225–256.

Orsvärn, K., 1994. Towards problem solving methods for sequential diagnosis. KADS-II/M2.3/TR/SICS/001/1.0, ESPRIT Project P5248.

de Prada, C., Alonso, C., Morilla, F., Bollain, M., 2001. Supervision and advance control in a beet factory. In: Biegler, L.T., Bramill, A., Scali, C., Marchetti, G. (Eds.), Advanced Control of Chemical Processes 2000 (ADCHEM 2000), vol. 1, Proceedings of the IFAC Symposium, Pisa, 14–16 June 2000. Elsevier Science, Oxford, pp. 341–346.

Pulido Junquera, B., Acosta, G., LlamasBello, C., Alonso González, C., 1998. TURBOLID: a KBS considering temporal information and performing diagnosis in a beet sugar factory. Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems. EIS'98, ICSC. Academic Press, New York.

Schreiber, G., Wielinga, B., Breuker, J., 1993. A Principled Approach to Knowledge-Based System Development. Academic Press, New York.

Steels, L., 1990. Components of expertise. AI Magazine, Summar, 29–49.

Tansley, D.S., Hayball, C.C., 1993. Knowledge-Based System analysis and Design—A KADS Developer's Handbook. Prentice-Hall, Englewood Cliffs, NJ.

Tzafestas, S.G., 1989. System fault diagnosis using the knowledge-based methodology. In: Fault Diagnosis in Dynamic Systems. Theory and Applications. Prentice-Hall, Englewood Cliffs, NJ (Chapter 15).

Gerardo Acosta graduated as Engineer in Electronics at the National University of La Plata, Argentina (1988), and as Ph.D. in Computer Science, in the Faculty of Sciences at the University of Valladolid, Spain (1995). Since then he has been Professor of Control Systems (Electronic Area) in the Faculty of Engineering at the National University of the Center of Buenos Aires Province (UNCPBA), at Olavarría, Argentina. He is a researcher of the National Research Council (CONICET). His working topic comprises the use of artificial intelligence techniques in control and supervision. He has more than forty papers in this and related fields. At present, he is also the Director of the Electromechanical Engineering Department and Co-director of the R + D Group "ADQDAT", at Faculty of Engineering-UNCPBA.

Carlos Alonso obtained his degree in Physics in 1985 and his Ph.D. in 1990 at the University of Valladolid, Spain. He has worked in several projects related to supervision and diagnosis of industrial plants. He is Associate Professor at *Departamento de Informática* of this University, and Secretary of the Spanish *Centro de Tecnología Azucarera* (Sugar Technology Center). His current research interests include topics such as Supervision and Diagnosis of Continuous Processes, Model-Based Diagnosis of Dynamic Systems, Knowledge Engineering and Machine Learning.

Belarmino Pulido graduated in Computer Science in 1992 in the Department of Computer Science, in the University of Valladolid, Spain, where he obtained his MSc in 1995, and Ph.D. degree in Computer Science in 2001. He has been an assistant professor in the University of Valladolid since 1995, and part-time researcher at the *Centro de Tecnología Azucarera* (Sugar Technology Center). His main research interests are: Model-based reasoning, Model-based Diagnosis, Consistency-based Diagnosis, and Knowledge-based Systems for Supervision and Diagnosis.