UNIVERSITAT POLITÈCNICA DE CATALUNYA

DEPARTAMENT DE LLENGUATGES i SISTEMES INFORMÀTICS

PROGRAMA DE DOCTORAT EN INTEL.LIGÈNCIA ARTIFICIAL

TESI DOCTORAL

DAI-DEPUR: AN INTEGRATED SUPERVISORY MULTI-LEVEL ARCHITECTURE FOR WASTEWATER TREATMENT PLANTS

Memòria presentada per en Miquel Sànchez i Marrè, per a optar al títol de Doctor en Informàtica per la Universitat Politècnica de Catalunya

Director: Dr. Ulises Cortés García

Barcelona, Hivern de 1995 / 1996

A l'Anna,
pel seu recolzament,
per la seva paciència,
i pel seu amor .

Contents

Resum

En aquesta tesi es presenta la recerca i el treball fet en el disseny i la implementació d'una *arquitectura integrada multi-nivell de supervisió* en temps real, de plantes de tractament o estacions depuradores d'aigües residuals (EDAR). La recerca desenvolupada recull una doble font de problemes: per una banda, la insuficiència dels mètodes habituals d'Enginyeria Química aplicats a les EDARs, i per l'altre, les limitacions dels Sistemes Basats en el Coneixement, en ésser enfrontats amb problemes del món real.

La gestió, control i supervisió d'una EDAR és una tasca molt complexa i perillosa, degut a les característiques de les EDARs i a les catastròfiques conseqüències a que pot donar lloc un mal funcionament de la planta. Les tècniques habituals –algorismes de *control numèric*– no són capaces de controlar la EDAR, tret de quan es troba en unes condicions de funcionament normals. Necessiten la *integració* d'altres tècniques que permetin incorporar el coneixement *expert* dels operadors de les plantes i dels llibres, així com les *experiències* adquirides en el funcionament passat de la planta sota control.

Els Sistemes Basats en el Coneixement (SBC) tenen una sèrie de mancances, sobretot quan s'apliquen a sistemes complexos del món real: no solen estar preparats per a fer front a situacions inesperades, la majoria dels SBC no aprenen de les seves experiències, hi ha greus dificultats per a l'adquisició del coneixement, les bases de coneixement solen ser poc reusables, i la complexitat creixent dels SBC monolítics.

L'arquitectura, anomenada DAI-DEPUR, es fruit de la *integració* de diverses tècniques d'Intel.ligència Artificial amb d'altres d'Enginyeria de Control, i amb altres d'Enginyeria Química: tècniques de control numèric –un algoritme de control predictiu–, models d'Enginyeria Química, raonament basat en el coneixement, raonament basat en casos, adquisició semiautomàtica de coneixements, aprenentatge, adquisició de dades on-line i off-line, *etc.*

L'objectiu global de supervisió es duu a terme de forma *distribuida* mitjançant una sèrie de tasques: evaluació del sistema, diagnosi local dels subsistemes, adaptació, diagnosi global, supervisió, validació i actuació. El coneixement expert està distribuit entre diferents bases de coneixement que cooperen per a la supervisió global de la planta. La arquitectura és *multi-nivell*, essent estructurada a partir de

l'estudi dels diferents tipus de coneixement i de les diferents tasques a realitzar. Aquest fet, doncs, proporciona una certa independència als diferents nivells que la composen: nivell de dades, nivell de coneixement/expertesa, nivell de situacions i nivell de plans d'actuació.

La implementació de DAI-DEPUR s'ha realitzat utilitzant certes eines com G2 –un shell per a sistemes experts en temps real–, LINNEO$^+$ –una eina no supervisada per a l'adquisició semiautomàtica de coneixements, GAR –un generador automàtic de regles de inferència–, i el llenguatge de programació Lisp per a la implementació d'un sistema de raonament basat en casos.

L'evaluació del sistema, que ha donat bons resultats, s'ha fet a dos nivells. Primer, s'han validat els tres grans components de l'arquitectura: el coneixement de control numèric, el coneixement expert i el coneixement experiencial. En segon lloc s'ha procedit a una validació global de DAI-DEPUR, que també consta de tres fases: simulacions del funcionament de la planta validades pels experts, validació en una planta pilot que s'ha construït a tal efecte, i la propera evaluació en una planta real, mitjançant un acord amb la Junta de Sanejament de la Generalitat de Catalunya.

Finalment es mostren varis aspectes de l'execució de DAI-DEPUR i es detallen uns quans exemples d'aplicació per a mostrar el procés global de supervisió de la planta on interactuen les diverses tècniques implementades, la EDAR i l'operador, a través de vàries interfícies.

Paraules Clau

Arquitectures Integrades, Arquitectures Multi-nivell, Arquitectures Distribuïdes, Raonament basat en el coneixement, Raonament basat en casos, Adquisició de coneixements, Aprenentage, Supervisió i Control en Temps Real, Tractament d'aigües residuals, Biotecnologia, Enginyeria Química, Enginyeria Medioambiental.

Abstract

In this thesis, it is presented the research and work developed in the design and implementation of *an integrated multi-level architecture for wastewater treatment plants (WWTPs) supervision* in real-time. The research has coped with a double open problems in two different areas: the insufficiency of classical Chemical Engineering control methods applied to WWTPs, and on the other hand, some pitfalls of Knowledge-Based Systems, specially when faced against real-world problems.

The management, control and supervision of a WWTP is a very complex and dangerous task, due to the features of a WWTP and to the catastrophic consequences that can be achieved by an incorrect WWTP operation. Usual used techniques –*numerical control* algorithms– are not able to control the WWTP if it is not operating in normal conditions. They need the *integration* of other techniques that allow to include the *expert knowledge* provided by the WWTP's operators and the literature, and the *experiential knowledge* acquired in the past operation of the WWTP under supervision.

Knowledge-Based Systems (KBS) have some pitfalls, specially when faced against complex real-world domains: their scope is limited to the forecasted situations in the domain, i.e. brittleness; most KBS do not learn from their experiences; the knowledge acquisition problem; low reusability of knowledge bases, and the increasing complexity of monolithic problem solving systems

The architecture, called DAI-DEPUR, is the result of the *integration* of several Artificial Intelligence techniques with some Control Engineering methods, and with some Chemical Engineering techniques: numerical control methods –a predictive control algorithm–, Chemical Engineering models, rule-based reasoning, case-based reasoning, semi automated knowledge acquisition, learning, on-line and off-line data acquisition, *etc.*

The global issue of supervision is carried out in a *distributed* way by means of several tasks: system evaluation, local diagnosis of subsystems, adaptation, global diagnosis, supervision, operator's validation and actuation. The expert knowledge is distributed among several knowledge bases that cooperate for the global supervisory task. The architecture is *multi-level*, and it has been structured in this way, as a result of the study of the different kinds of knowledge and tasks involved

in the domain. This feature provides it with a certain independence among the different levels: the data level, the knowledge/expertise level, the situations level and the plans level.

DAI-DEPUR has been implemented by means of some tools such as G2 – a real-time expert systems shell–, LINNEO$^+$ –a semi-automated unsupervised knowledge acquisition tool–, GAR –an inference rule automated generator–, and the programming language Lisp for the implementation of the case-based reasoner.

The evaluation of the system has given good results and it has been carried out in two stages. First, the three main components of the architecture: the numerical control knowledge, the expert knowledge and the experiential knowledge. Secondly, a global validation of DAI-DEPUR, also containing three steps has followed: WWTP operation simulations validated by the experts, validation in a pilot scale WWTP constructed to that end, and the next evaluation in a real WWTP by means of an agreement with the "Junta de Sanejament de la Generalitat de Catalunya".

Finally, some features of DAI-DEPUR execution are showed, and a few examples of application are detailed, in order to outline the global supervisory process where interact the several techniques implemented, the WWTP and the WWTP's operator through several interfaces.

Key Words

Integrated Architectures, Multi-level Architectures, Distributed Architectures, Rule-based reasoning, Case-based reasoning, Knowledge acquisition, Learning, Real-time Supervision and Control, Wastewater treatment, Biotechnology, Chemical Engineering, Environmental Engineering.

Agraïments

Voldria compensar d'alguna forma a tothom, que d'una manera o altra, m'ha ajudat en la realització d'aquesta tesi. En la mena de peregrinació que ha suposat aquesta tesi des de que vaig acabar la carrera, hi ha passat de tot. Primer, va haver-hi el servei militar. Després la Tesi de Llicenciatura que va apuntar-ne el camí. Després, fer de professor a la universitat amb la càrrega docent que comporta també ha tingut el seu pes, i per si això no fos poc, el estar sotmès a la constant pressió –científica– d'en Ulises.

Les idees i la recerca exposada en la tesi són el resultat de converses, discussions i intercanvis d'impressions amb molta gent. Principalment amb el meu director, en Ulises Cortés, que ja fa anys em va introduir en el món de la Intel.ligència Artificial, en el de la Universitat i en el de la ciència. Sempre ha estat disposat –encara que no tingués gaire temps lliure– a ajudar-me i a apuntalar la tesi. Gràcies per aquests anys, d'amistat i col.laboració.

Hi ha un grup de gent que també m'han ajudat molt en la tesi, i que al llarg d'aquests anys ens hem anat coneixent –i compartint deu n'hi do quants àpats gastronòmico-científics– fins a arribar a ser força amics. Són la gent de la Unitat d'Enginyeria Química de la Universitat Autonòma de Barcelona i del Laboratori d'Enginyeria Química i Ambiental de la Universitat de Girona. Ells són els "culpables" de la meva dedicació als temes ambientals, i a les plantes depuradores en concret. Han sigut molts en aquests anys: En Jordi Robusté, en Pau Serra, en Joan de Gràcia, en Juan Baeza que ha posat a punt l'adquisició de dades on-line, en Paco, en David. Capítol apart són en Javier Lafuente, en Manel Poch i en Ignasi R.-Roda. En Javier et contagia de la seva energia vital i de la seva disposició a qualsevol hora, i en Manel de la seva visió científica i la seva afecció pels viatges a México. Gràcies a tots dos pels seus consells, crítiques constructives i idees. Que dir d'en Ignasi. S'ha volcat en ajudar-me. Són molts dies del seu temps (t'en recordes del 24/1/96? Aquell si que va ser un dia profitós!) que m'ha dedicat generosament. Gràcies pel teu esforç. Sense tu no ho hagués aconseguit.

També m'han ajudat molt els companys i ex-companys del grup de Sistemes Basats en el Coneixement i Aprenentage, especialment en Javier Béjar, sempre amb la solució a la mà per a qualsevol problema del computador i per haver construït la versió actual de LINNEO$^+$, que ha sigut essencial per la tesi. En Toni Moreno també ha col.laborat, sobre tot animant-me quan estava més decaigut, discutint certes idees i empenyent-me a escriure-la en anglès. A en Julio Valdés, que amb la seva amabilitat i la seva ciència també ha aportat el seu granet de sorra per ajudar-me. A en David Riaño, per haver construït GAR. A en Juan Manuel Gimeno, a en Lluís Belanche, a en Enric Sesa, a en Mario Martín, que també m'han ajudat durant aquest temps.

A en Enric Plaza, que va comentar diferents aspectes i l'enfoc de la tesi, i va corregir certes parts, sense saber-ho. A tota la gent del IIIA que hem el seu dinamisme i exemple científic t'estimulen constantment.

Als nostres experts: a el Sr. Canals, ja fa força temps; a en Ricard Tomàs, cap de la planta de Manresa-Sant Joan de Vilatorrada; a en J. Torrico, cap de la planta de Cassà de la Selva-Llagostera. A en Javier Lafuente, Manel Poch i Ignasi R.-Roda que també en són molt d'experts.

A tota la gent del departament de LSI, que d'una forma o altre ens anem fent costat en aquestes doloroses peregrinacions. I al departament en abstracte, pel suport informàtic que ens ofereix i per la descàrrega docent que em van concedir. En particular a la secció de IA, per haver-me concedit un parell d'ajuts per a finançar part de les despeses en les anades a congressos.

A la família: Anna, Paquita i Salvador, Carme i Sisquet, Maria Reina, Carme i Josep Maria, Laura, i a la resta pel seu suport afectuós, i pel temps que no els he pogut dedicar com jo hagués volgut durant aquest periode, especialment en l'etapa final de la tesi.

Als amics i altres companys, que sigui com sigui, m'han animat en aquesta tasca.

Aquesta tesi ha rebut el suport econòmic dels següents projectes científics per part de les institucions:

- "Un paquete inteligente de diseño asistido por ordenador de sistemas de control de procesos biotecnológicos". CICyT ROB89-0479-C03-02.
- "VIM: a Virtual Multicomputer for Symbolic Applications". EEC. ERB4050PL930186.
- "Desarrollo de un sistema basado en el conocimiento para la supervisión y el (re)diseño de procesos biotecnológicos: aplicación a un proceso biológico de depuración". CICyT BIO94-679-C02-01.

Preface

The idea of using Artificial Intelligence techniques, particularly Expert Systems, to the management and supervision of wastewater treatment plants was born in the Chemical Engineering Unit of the UAB. This approach joined with the efforts of the Knowledge-Based Systems and Machine Learning group of the Artificial Intelligence section of the Software department (LSI) of the UPC, to overcome some troubles in knowledge engineering and knowledge acquisition. The merge of both paths has derived to this thesis.

The work was based on my early Master's Thesis [Sànchez, 1991] that was focused on the use of Knowledge-Based Systems for the off-line diagnosis task in wastewater treatment plants. That work was awarded with an accesit of the "Oms i De Prat 1991" award in the modality of Experimental and Applied Science. The KBS system developed was called DEPUR. DEPUR stands for DEPURation KBS system.

From this experience and joining the efforts of all groups involved in the research: chemical engineers, control engineers, biologists, plant's operators –experts–, the architecture for the global supervision, control and management of the wastewater treatment plant was designed and afterwards implemented. It has been called DAI-DEPUR, that stands for Distributed And Integrated DEPUR.

As the work is the result of this collaboration, the thesis is an interdisciplinary research. Although this interdisciplinarity, the work is presented here from an Artificial Intelligence point of view. It has been a challenge to produce a text that could be meaningful and useful to both types of reader, although it is mainly addressed to Artificial Intelligent scientists. So, it is possible that some topics are not fully explained here, because they are mainly related to the other discipline. Anyway, a glossary of technical words in Chemical Engineering is provided in the appendix A.

One of the goals of our research –and that was clearly pointed by the experts in the early stage– was that the supervisory architecture to be designed would be general enough to be used in any wastewater treatment plant of similar technology. The proposed architecture satisfies this requirement, because it can adapt itself through a dynamic experiential component.

Although the work has focused on the wastewater treatment plants, we aim that the architecture is also useful for other real-time control processes in complex real-world domains.

The text is written in English due to the aim of providing an easy understanding and a wide dissemination of the work to the interested scientific community, if any. Writing the text in a foreign language has supposed an added difficulty to the work, and a rejection of my own loved Catalan language. As it is not written by an Oxford scholar, but a foreign writer, be benevolent with my English grammar and language mistakes !

List of Figures

List of Tables

Chapter 1

Introduction

What has been now started as a science fiction novel, tomorrow will be finished as a report.
Arthur C. Clarke

1.1 Motivations

The primary objective of wastewater treatment plant (WWTP) operation is to meet the specified requirements on the outflow water quality, in order to restore the natural environmental balance broken by human beings activities (industrial wastes, domestic waters, *etc*.). The process carried out in WWTP to accomplish that goal is very *crucial*, so that a bad outflow water quality is very dangerous for the human beings and the nature. Also, the process itself is very *complex*. Both of these features make difficult to set a reliable supervisory control technology over the wastewater treatment plants. On the other hand, Knowledge-Based Systems (KBS)– one of the most broadly applied paradigm of Artificial Intelligence– cope with some pitfalls, specially when faced against real world domains.

Thus, the motivations of this work were originated by a double source of unsolved problems: the proved insufficiency of Chemical Engineering classical control methods applied to WWTP supervision, and on the other hand, some limitations of Knowledge-Based Systems –specially in real world domains–, in Artificial Intelligence.

1.1.1 Shortcomings in classical process control methods

The setting of an automatic process control over the wastewater treatment plant system, has showed some difficulties:

- The *complexity of the system* There are many factors influencing the process, such as the complex interrelations between the microorganisms coexisting in the reactors and between these microorganisms and the substrate to be broken down. Moreover, this substrate is also affected by the variable quantity and quality of the wastewater stream, as illustrated in figure 1.1.

Fig. 1.1. Evolution of Biological Oxygen Demand (BOD) –a measure of the organic matter– along the time at the input of a WWTP

- An *ill-structured domain*. The relationships among the concepts or attributes of the domain are not enough known and there is not much agreement among the experts. The relationship among the various phenomena which characterize the system is insufficiently understood. Although different mathematical models have been put forward to describe the relationship between the microorganisms and the substrate, these models cannot be said to provide an entirely satisfactory description of the cause-effect relationships occurring within the plant.

- *Non-numerical or qualitative information*. There are many data related to the process which cannot be numerically quantified, and therefore cannot be used in the context of a conventional control model. For example, water appearance, water smell, microbiological information, state of the flocculation during sedimentation, *etc.*.

- *Uncertainty or approximate knowledge*. The variables which describe the process are global, such as the chemical or biochemical oxygen demand, the volatile solids, *etc.*, and difficult to obtain on-line. Therefore, the expert also has to take into account subjective information, based on local experience, which enables him to identify certain states in the plant.

- A *dynamic system*. The system is under continuous changes that can directly modify the performance of the process. Therefore, a real time control loop is needed to supervise the process.

Moreover, it can be said that the *classical control methods* work well on the *normal* states of the plant but not in other *abnormal* states of working:

- How could they detect some unforeseen situations such as the mechanical faults or cope with a toxic substances shock ?
- How could they use the subjective information accumulated through years of experience by the experts ?
- How could they use the available but incomplete information, to solve an specific problem ?
- How could they use the objective information provided by years of WWTP operation?

1.1.2 Limitations of Knowledge-Based Systems

The issue of Knowledge-Based Systems (KBS) is the emulation of human problem solving capabilities, using the same knowledge sources, within a concrete domain [González and Dankel, 1994; Jackson, 1990; Buchanan and Smith, 1988; Clancey, 1985a; Hayes-Roth, 1984; Stefik *et al.*, 1982]. Knowledge-based systems are formed of a set of either declarative or procedural informations and relationships. Also, they have certain heuristics that form the *knowledge body*, and some inference and search processes. Main problems solved with knowledge-based systems are usually solved by human experts and considered as very complex and specialized ones.

Usually, a great amount of knowledge is required to solve these kind of problems. Typically, the *knowledge body* is encoded in form of inference rules that allows the system to deduce some new conclusions, from a set of premises or data:

IF <conditions> THEN <actions>

Commonly, the reasoning method (inference engines) may be forward chaining, backward chaining or a combination of both. Forward reasoning is guided from the input data to the conclusions of the system, by means of deducing some new facts from previous ones. In backward chaining, the reasoning is guided from the conclusion(s) –that is(are) wanted to deduce– to the input data provided by users.

Main components of KBS are: the knowledge base or long-time memory, the data base or working memory (short-time memory), the inference engines, the user interface, the auto-explanation module, the strategies or control module, the knowledge engineer interface, on-line sensors/actuators interface, *etc.* (see figure 1.2). In the more advanced generations of KBS such as in [Puyol, 1994], the modules of inference rules are considered as *specialists* in a given subdomain.

The main characteristics of Knowledge-Based Systems (KBS) point to the fact that they could be used for the supervision and control [Sànchez *et al.*, 1995c; Sànchez *et al.*, 1994a; Dym and Levitt, 1991; Stephanopoulos, 1990; Sriram and Adey, 1987; Efstathiou and Mamdani, 1985]. In our case, we want to focus on the supervision and control of real wastewater treatments plants. Among those characteristics we can distinguish:

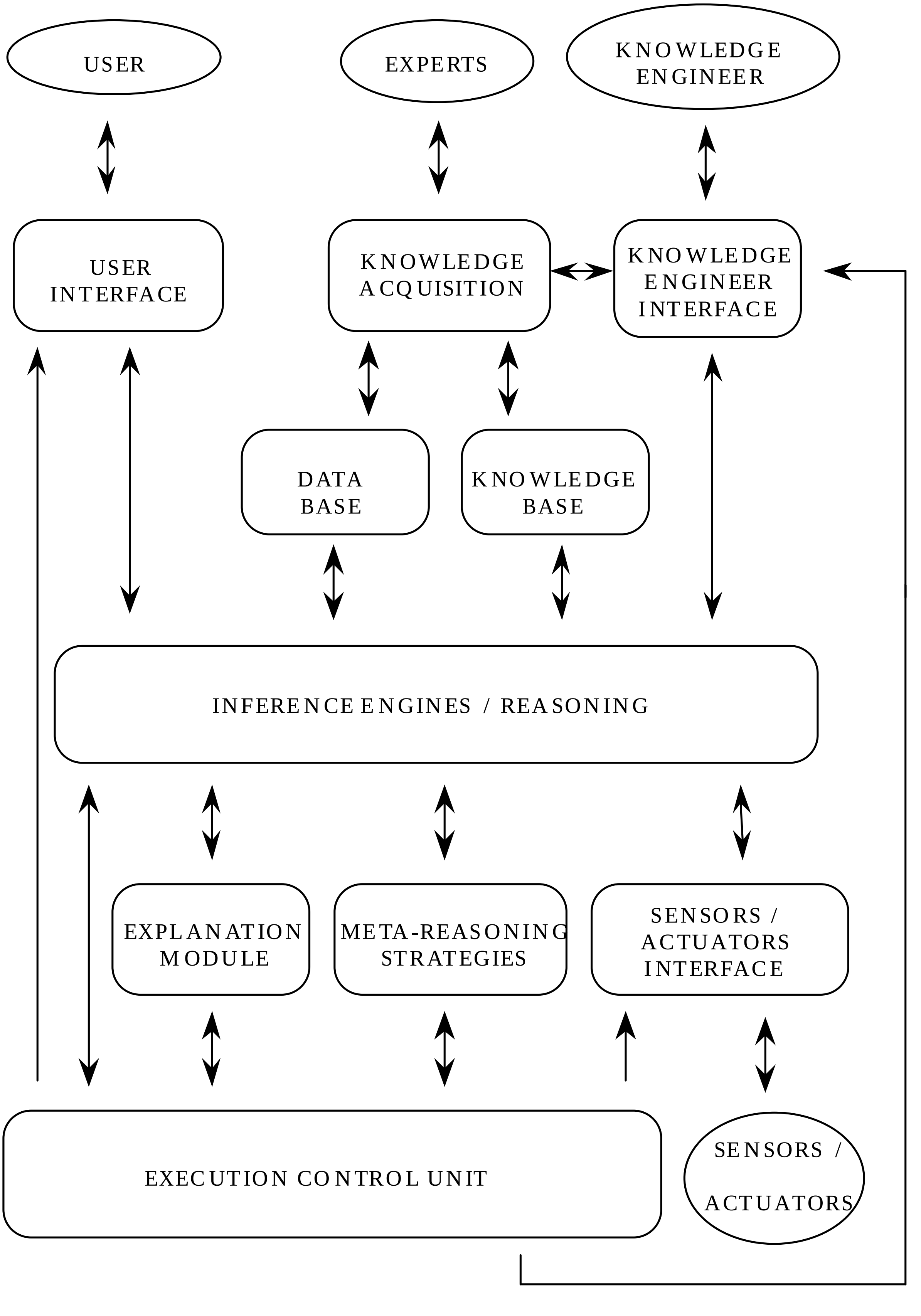


Fig. 1.2. Knowledge-Based System

- *Usefulness in concrete domains*. They are effective when applied to a certain domain where some expert people can afford their experience and knowledge.
- Separation of knowledge base from control elements. It provides a more easily way of building and updating knowledge-based systems.
- Getting the *expertise and experience from human beings* within a specific domain into the system.
- High interactive systems.
- *Natural language interfaces* are usually provided for its exploitation.
- Supporting of *numerical and/or symbolic information*.
- Specially useful in *ill-structured domains*.
- Could be extended in some ways. For example, the treatment of *uncertain or approximate reasoning*.

Some of the problems within the conventional process control systems have been the focus during the last years, of much of the research efforts in Artificial Intelligence –specially in KBS– applied to real world problems:

- Control of petrochemical plants, [Alamán *et al.*, 1992].
- Monitoring of continuous processes, [Finch *et al.*, 1990].
- Statistical process control, [Anderson *et al.*, 1990; Novotny *et al.*, 1990].
- Control of sun powered plants, [Sanz *et al.*, 1989].
- Real time process control, [Wright *et al.*, 1987; Moore *et al.*, 1984].

Nevertheless, Knowledge-Based Systems do not incorporate some desired features from human intelligence and have some technical difficulties in their development:

- Most KBS *do not learn from their experiences*. The use of experience is a valuable feature to be contemplated in KBS [Aamodt, 1989].

- *The knowledge acquisition problem*. There are some difficulties in the process of extracting the knowledge and experience from knowledge's sources. [Becker, 1987].

- *Low reusability of knowledge bases*. Knowledge acquisition strongly depends on both the experts and the concrete domain. Thus, it is very difficult the partial or global sharing and reuse of knowledge bases [Neches *et al.*, 1991].

- *Brittleness.* Their scope is limited to the forecasted situations in the domain. They are not reliable when applied to unexpected situations [Steels, 1990].

- *The increasing complexity of monolithic problem solving systems.* As the systems grow, it is more difficult to manage the information and knowledge contained in them.

1.2 Wastewater Treatment Plants Process

1.2.1 Wastewater

New big urban areas produce a big amount of wastewaters. When the natural environmental balance is broken, the water quality is getting worse due to the fact that these wastewaters overcome the performance of auto-regulation process of the receiving waters. In this case, society has to act in order to restore this natural environmental balance.

Municipal wastewater treatment plants (WWTP) provide an important buffer between the natural environment and the concentrated wastewaters from urban areas. If released in an uncontrolled fashion, these wastewaters would degrade the water, land and air on which life depends.

If untreated wastewater is allowed to accumulate, the decomposition of the organic materials that it contains can lead to the uncontrolled production of large quantities of malodorous gases. In addition, untreated wastewater usually contains numerous pathogenic, or disease-causing, microorganisms that dwell in the human intestinal tract or that may be present in certain industrial waste. Wastewater also contains nutrients, which can stimulate the growth of aquatic plants, and it may contain toxic compounds. For these reasons, the immediate and nuisance-free removal of wastewater from its sources of generation, followed by treatment and disposal, is not only desirable but also compulsory in an industrialized society, [Metcalf & Eddy, 1991; Benefield and Randall, 1980].

The wastewater quality and quantity characteristics of a plant's influent, typically, reflect the nature of the contributing area, water uses, and conditions of the conveyance system. From the standpoint of sources of generation, *wastewater* may be defined as a combination of the liquid –or water– carrying wastes removed from residences, institutions, commercial and industrial establishments; together with that groundwater, surface water and stormwater may be present too. Typical data on the individual constituents found in domestic wastewater are reported in Table 1.1. Both the constituents and the concentrations vary with the hour of the day, the day of the week, the month of the year and, as mentioned above, with other local conditions.

Contaminants	Concentration (mg/l)
SOLIDS	
Dissolved	250-850
Suspended	100-350
Settleable	5-20 ml/l
ORGANIC MATERIA	
BOD_5	110-400
TOC	80-290
COD	250-1000
NUTRIENTS	
Nitrogen (total as N)	20-85
organic	8-35
free amonia	12-50
nitrates	0
nitrites	0
Phosphorus (total as P)	4-15
organic	1-5
inorganic	3-10
PATHOGENS	
Total coliform	10^7-10^{10}

Table 1.1. Typical composition of untreated domestic wastewater

The main objective of wastewater treatment plants operation is to meet the specified requirements or, if the facility is nondischarging, the applicable requirements of the regulatory agencies for groundwater protection (see appendix B for the Catalonian regulations). At the same time, the operation must protect the safety, health, and well-being of the plants' employees and neighbours. In establishing the requirements for wastewater treatment, the regulatory agencies may consider the following, as well as compliance with minimum statutory requirements, [WPCF, 1990]:

- Prevention of disease

- Prevention of nuisances
- Avoidance of water supply contamination
- Elimination of all pollutant discharges to navigable waters
- Maintaining clean waters for the propagation and survival of fish and other aquatic life
- Protection of waters for personal bathing and recreational use
- Preservation of pristine waters for ecosystem protection
- Conservation of water

1.2.2 Waste water treatment

Unit operations, i.e. treatment methods in which the application of physical forces predominates, and unit processes, i.e. treatment methods in which the removal of contaminants is brought about by chemical or biological reactions are grouped together to provide what is known as preliminary, primary, secondary and advanced or tertiary treatment, if it exists. A chart of this process is depicted in figure 1.3.

1.2.2.1 Preliminary treatment

Preliminary treatment of wastewaters may include screening, grit removal, chemical additions, pre-aeration, odour control (where appropriate) and flow measurement. The removal of debris in the screening area and the removal of sand, rocks, gravel and other inorganic in the grit removal system protect downstream treatment processes.

1.2.2.2 Primary treatment

Primary treatment is normally associated with sedimentation but occasionally uses fine screens. Primary sedimentation separates the readily settleable and floatable solids from the wastewater for subsequent wastewater treatment. Other benefits of primary settling include equalization of sidestream flows and removal of the biological oxygen demand (BOD) associated with settleable solids.
Many treatment plants use primary sedimentation tanks for thickening primary sludge as well as solids separation from wastewater. Well-designed and operated primary treatment facilities may remove as much as 60 to 75 % of the influent suspended solids and up to 35 % of the biodegradable organic material.

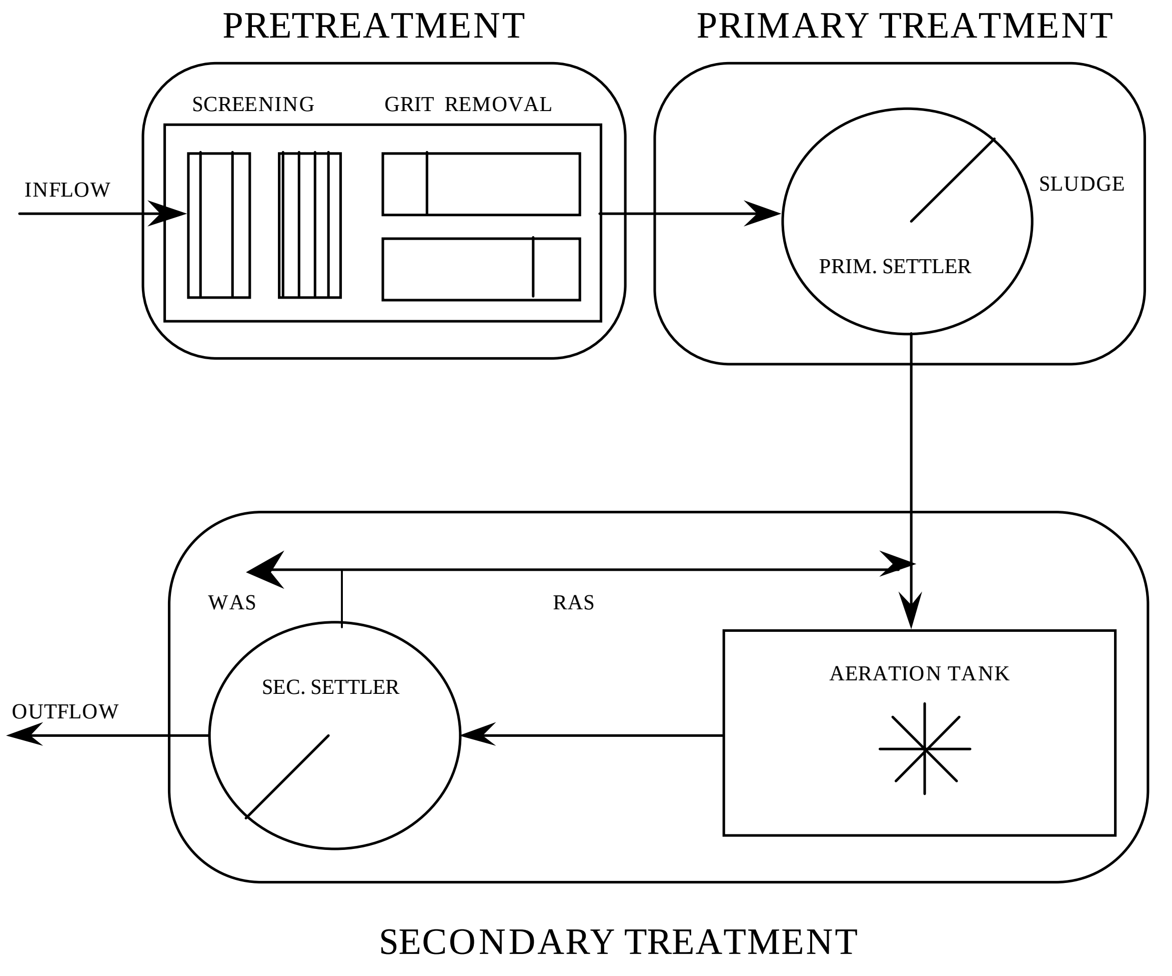


Fig. 1.3. Chart of a wastewater treatment plant

1.2.2.3 Secondary treatment

Most secondary treatment processes involve biological treatment, using metabolic reactions of microorganisms to produce a high quality effluent by converting and removing substances that have an oxygen demand. Most of these biological processes can be classified as *attached growth* such as trickling filters, packed towers and rotating biological contactors, or *suspended growth systems* referred to as activated sludge.

In the basic *activated sludge process*, [Robusté, 1990], the wastewater goes into an aerated tank where previously developed biological floc particles are brought into contact with the organic matter of the wastewater. The organic matter, a carbon and energy source for cell growth, is converted into cell tissue and oxidized end products (mainly carbon dioxide, CO_2). The contents of the aeration tank are called

mixed liquor. The biological mass, referred to as the mixed liquor suspended solids (MLSS) or mixed liquor volatile suspended solids (MLVSS), consists mostly of microorganisms, inert suspended matter and non biodegradable suspended matter.
After the mixed liquor is discharged from the aeration tank, a clarifier that also is referred to as settling tank or sedimentation tank separates the suspended solids (SS) from the treated wastewater. The concentrated biological solids are then recycled back to the aeration tank to maintain a concentrated population of microorganisms to treat the wastewater. Because microorganisms are continually produced, i.e. synthesized, in this process, a way must be provided to waste excess biological solids produced. These solids are generally withdrawn from the clarifier, although wasting from the aeration tank is an alternative.
Activated sludge is the most vital wastewater treatment process today. For almost a century, it has been successfully utilized as a conventional system for carbon removal. For the last few decades, its potential to remove nutrients, such as nitrogen or phosphorus, has been explored and tested. Despite substantial practice and accumulate experience, it is wide open to research and conceptual development.

In *attached growth systems*, the mass of microorganisms affecting treatment are attached to supporting media as a biological film. This film, a viscous jelly-like slime, generally comprises a large and diverse population of living organisms, including bacteria, protozoa, algae, fungi, worms, and even insect larvae. Of the total mass of the diverse population, most of the mass consists of microorganisms that consume food and require oxygen to remain aerobic.
The removal of soluble organic material is a relatively rapid process. Good removal of soluble organic material can generally be achieved at low to moderate loading of the fixed-film reactors. However, the stabilization or breakdown of biological solids generated in removing the soluble organic material is a longer process. The time required for completion of this process will vary depending on the type of filter media being used, rate of organic loading to the fixed-film reactor, hydraulic shear, temperature, and other factors.

1.2.2.4 Advanced treatment

Advanced wastewater treatment may be used to reduce the concentrations of nutrients, nitrogen or phosphorus, and soluble organic substances to levels below those normally attained through secondary treatment.
Basically, the processes developed can be considered as modifications of conventional activated sludge technology. The major differences are related to the

provision for contrasting environmental conditions in the bioreactor and careful control of the solids retention time (SRT). This encourages the selective growth of certain types of microorganisms in the reactor's biomass. An advanced treatment may also include ozonation, carbon ascription, ion exchange, reverse osmosis, electrodialysis, *etc.* [WPCF, 1990; Culp and Culp, 1978].

1.2.3 Wastewater operation and control

The control of the biological process is important to maintain high levels of treatment performance under a wide range of operating conditions. Successful process control consists of reviewing present and historical operating data and laboratory test results to select the proper operational parameters that provide the best performance at the least cost. Focusing on activated sludge configuration, the principal factors used in process control are:

- Maintaining dissolved-oxygen (DO) levels in the aeration tanks
- Regulating the amount of return activated sludge (RAS)
- Controlling the waste activated sludge (WAS)
- If necessary, chemical feed rates such as chlorine, settling aids, nutrients, *etc.*

1.2.3.1 Aeration and Dissolved Oxygen control method

The purpose of aeration is two-fold: oxygen must be dissolved in the liquid in sufficient quantities to maintain active organisms, and the contents of the tanks must be sufficiently mixed to keep the sludge solids in suspension and uniformly mixed with the wastewater.
The amount of oxygen that must be transferred in the aeration tanks theoretically equals the amount of oxygen required by the microorganisms in the system to oxidize the organic material. However, excess oxygen must be supplied to maintain the Dissolved Oxygen (DO) at the centre of the floc particle and to sustain the desirable microorganisms in clarifier and return sludge line back to the aeration tank. When oxygen limits the growth of microorganisms, filamentous organisms may predominate, and the settleability and quality of the activated sludge may deteriorate. On the other hand, over-aeration wastes energy, may create excess turbulence, and may break up the biological floc resulting in poor settling and high effluent solids. In practice, the dissolved-oxygen concentration in the aeration tank

should be maintained at about 1,5 to 4 mg/L in all areas of the aeration tank; 2 mg/L is a commonly used value.

1.2.3.2 Return of Activated Sludge control methods

Return of Activated Sludge (RAS) is a key control parameter of the process. The purpose of the return of the MLSS settled in the clarifier is to maintain a sufficient concentration of activated sludge in the aeration tank so that the required degree of treatment can be obtained in the time interval desired. The rate at which the activated sludge is returned from the final clarifier to the inlet of the aeration tank affects the solids balance between these units. There are three basic ways for returning sludge to the aeration tank:

- At a constant rate, independent of the secondary influent flow rate; it results in a continuously varying MLSS concentration. Therefore, the depth of sludge blanket in the clarifier constantly changes as the MLSS moves from the aeration tank to the clarifier and vice versa.

- At a constant percentage of the varying secondary influent flow. This approach keeps the MLSS and sludge blanket depths more constant throughout high and low flow periods and also tends to maintain a more constant food-to-microorganism ratio (F/M), and sludge retention time (SRT; average number of days that microorganisms are kept in the activated sludge process).

- At a varying rate to optimize the concentration and detention time of the clarifier solids.

Several techniques are used to calculate the desirable return-sludge flow rate: settleability, direct sludge blanket level control, secondary clarifier mass balance, aeration tank mass balance or sludge quality. All these techniques are fundamentally similar; they differ on how accurately they control the solids inventory during abnormal conditions, when is more necessary an accurate process control.

1.2.3.3 Waste of Activated Sludge control methods

The most important technique used to control the activated sludge process is to control the solids inventory in the system with the wasting rate of the excess activated sludge produced each day. The Waste of Activated Sludge (WAS) removed from the process affects the effluent quality, the growth rate and the types of the present microorganisms, oxygen consumption, mixed liquor settleability, nutrient quantities needed, the occurrence of *foaming* and *frothing*, and the possibility of *nitrifying*.

The most common practice is to waste sludge from the return sludge line because it is more concentrated and requires smaller waste sludge pumps, but, as an alternative activated sludge wasting method, mixed liquor can be removed from the aeration tank.

The most common methods to control the amount of sludge wasted are:

- Constant SRT, widely used and reliable, particularly when various process control measurements are used in choosing the best SRT

- Constant F/M ratio. This method requires a significant amount of laboratory work because the incoming load must be determined

- Constant MLSS, with a minimum amount of laboratory control

- Sludge quality, that can be used independently or with other control methods, and includes laboratory tests and observations as microscopic examination of MLSS, aeration tank and secondary clarifier observations, effluent clarity, etc.

1.3 Issues

The main goal of a wastewater treatment plant is to reduce the pollution level of the wastewater at the lowest cost, that is, to remove –within the possible measure– strange compounds (pollutants) of the inflow water to the plant prior to discharge to the environment. So, the effluent water has the lower levels of pollutants as possible (in any case, lower than the maximum ones allowed by the law). The plants taken as models –in this study– are based on the main biological technology usually applied: *the activated sludge* process [Robusté, 1990]. The main target wastewater plant studied is located in Manresa-Sant Joan de Vilatorrada, near Barcelona (Catalonia). This plant receives about 30000 m^3/day inflow from 75000 inhabitants (see figure 1.4).

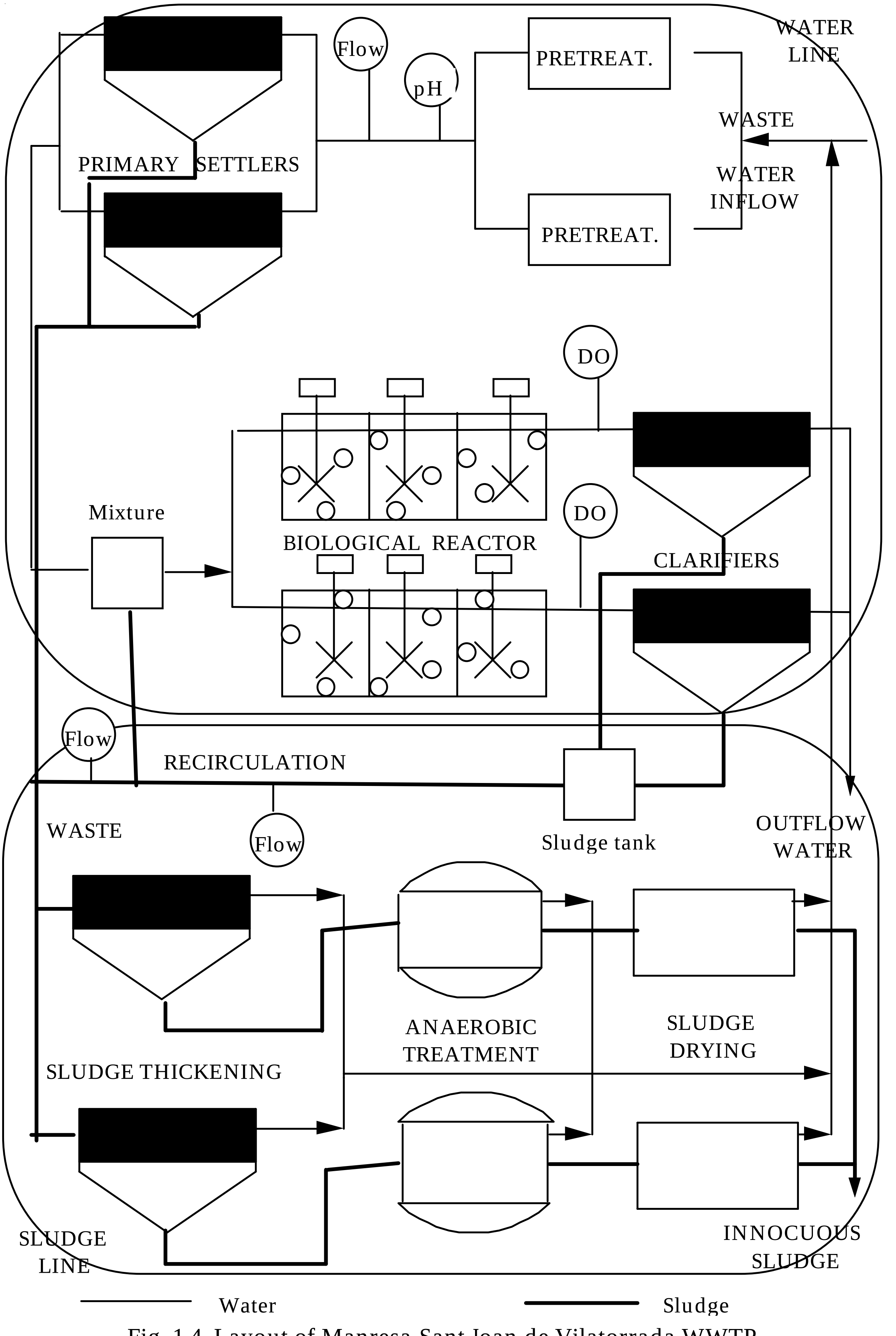


Fig. 1.4. Layout of Manresa-Sant Joan de Vilatorrada WWTP

Other wastewater treatment plant studied with a similar, but slightly different technology is located in Cassà de la Selva-Llagostera, near Girona (Catalonia). That plant receives about 2500 m^3/ day inflow from 10000 inhabitants.

The activated sludge process directly depends on live beings (microorganisms), and therefore, on changes experimented by them. It could be possible to get a good plant operation if the supervisory control system is able to react to the changes and deviations of the system and can take the necessary actions to restore the system's performance.

From what has been outlined in previous sections 1.1 and 1.2, it is clear that there are many operations of different nature meeting in a WWTP: mechanical, electrical, chemical, biological, microbiological, physical operations, *etc.* All of these operations can originate themselves failures which can lead the plant to a bad operation state, i.e. a bad outflow water quality. In addition to this complexity, there are some features that make very difficult the setting of a conventional numerical automatic control method such as feedback, feedforward, optimal, predictive control, *etc.* to manage the whole process. Namely, a high variability of the quantity and quality of the inflow water; the biochemical processes involved are not well-enough known; there is a lot of qualitative information; some on-line measures are usually not reliable; there is much subjective, useful, and uncertain information accumulated by the experience of the plants' experts; there are not two equal wastewater treatment plants; and, also, it is necessary to consider the seasonal effects on all these factors.

Control, supervision and the overall management of WWTP cannot be implemented within a single approach. It is needed a multi-disciplinary integrated way [Venkatasubramanian, 1994], that includes: *monitoring* (sensor developing, continuous analysis equipment), *modelling* (equations that model the bioreactors' behaviour), *numerical control* (maintaining good effluent water quality and reducing operation costs), *qualitative information* (microbiological information, water's colour and odour, water's appearance, *etc.*), *expert knowledge* (supplied by the large experience from plants' managers, biologists and operators) and *experiential knowledge* (specific knowledge supplied by the previous solved problems in the system). The three last features commonly provide the systems with incomplete, uncertain or approximate information. Therefore, AI can play a good role in WWTP supervision [Sànchez *et al.*, 1995c; Patry and Chapman, 1989; Hushon, 1987; Stephanopoulos and Stephanopoulos, 1986; Horan and Eccles, 1986].

On the other hand, one of the key problems in real-time Knowledge-Based control systems design is the development of an architecture able to manage efficiently the different elements of the process (*an integrated architecture*), to learn from past experience and to acquire the domain knowledge. These problems increase when the process is composed by several complex operational units. Therefore, a *distributed problem solving* AI architecture seems to be a good choice. On the other hand, it is generally agreed that more powerful *knowledge acquisition* tools and techniques are needed in order to increase both the quality and the quantity of Knowledge-Based Systems for real world applications. By them, we mean systems that exhibit a certain level of complexity, that sometimes have to cope with problems on the border (or slightly outside) of their special domain of competence (*not brittle*), and have to be properly self-updated and maintained (*learning*) in order not to degrade over time.

Taking into account all these limitations and the needs for a reliable real-time supervisory system, it is reasonable to think that an integrated and distributed problem solving architecture could be a new interesting approach to wastewater treatment plants management and control.

1.4 Scheme of the thesis

Chapter 1 is an introduction to the problems we are studying. In this chapter, there are explained the reasons that caused this research. First, the insufficiency of classical process control methods applied to wastewater treatment plants, and on the other hand, some limitations of knowledge-based systems applied to real-world systems. Background on wastewater treatments plants process is introduced by means of a brief survey on this topic. Finally, the issues of the work are presented and the organization of the work is detailed.

Chapter 2 shows the state of the art in the field of automatic process control applied to WWTP, and describes the main characteristics of real-time systems. It also gives a summary on Artificial Intelligence –and specifically in Knowledge-Based systems– applied to WWTP control. First, the knowledge-level theory interpretation among several authors to model a specific domain is discussed. Research on integrated cognitive architectures and multi-level architectures are detailed, too. Finally, an insight of distributed Artificial Intelligence is presented.

Chapter 3 presents the designed architecture: DAI-DEPUR. It is an integrated supervisory multi-level one. First, the knowledge level analysis of WWTP domain is discussed. Next, all the components and features of the architecture are exposed over the sections of the chapter: cognitive integration, multi-level architecture and distributed problem solving. Next chapters analyse the levels of the architecture through the knowledge level theory proposed by L. Steels: domain models, tasks and methods.

Chapter 4 explains the first level of the architecture: the data level. The domain theory is modelled by the on-line data, the calculated and inferred data, and off-line data, acquired through both interfaces: user interface and external interface. Main task at the level is the system evaluation which is implemented by the data gathering method.

Chapter 5 describes the second architecture level: the knowledge / expertise level. In this level, it is very important the cognitive integration of several paradigms involved. Domain theory is modelled by numerical control knowledge, by expert general knowledge and by experiential specific knowledge. Main tasks performed at this level are the diagnosis and the adaptation of the system to the real-world evolving process. Methods implemented to perform the tasks are the expert knowledge acquisition, learning from observation, the rule-based reasoning, the case-based reasoning, learning from experience and introspection.

Chapter 6 exposes the third level of the architecture: the situations level. Domain theory is represented by means of specific and generic working situations. The task at this level is the supervision of the WWTP that is performed by the combination method.

Chapter 7 details the upper level of the architecture: the plans level. Domain theory is modelled by identified situations and plans (solutions). Main tasks are the plan validation and the final actuation over the WWTP. The methods that perform these tasks are the experiential / expert actuation (canned plans) and the numerical control actuation.

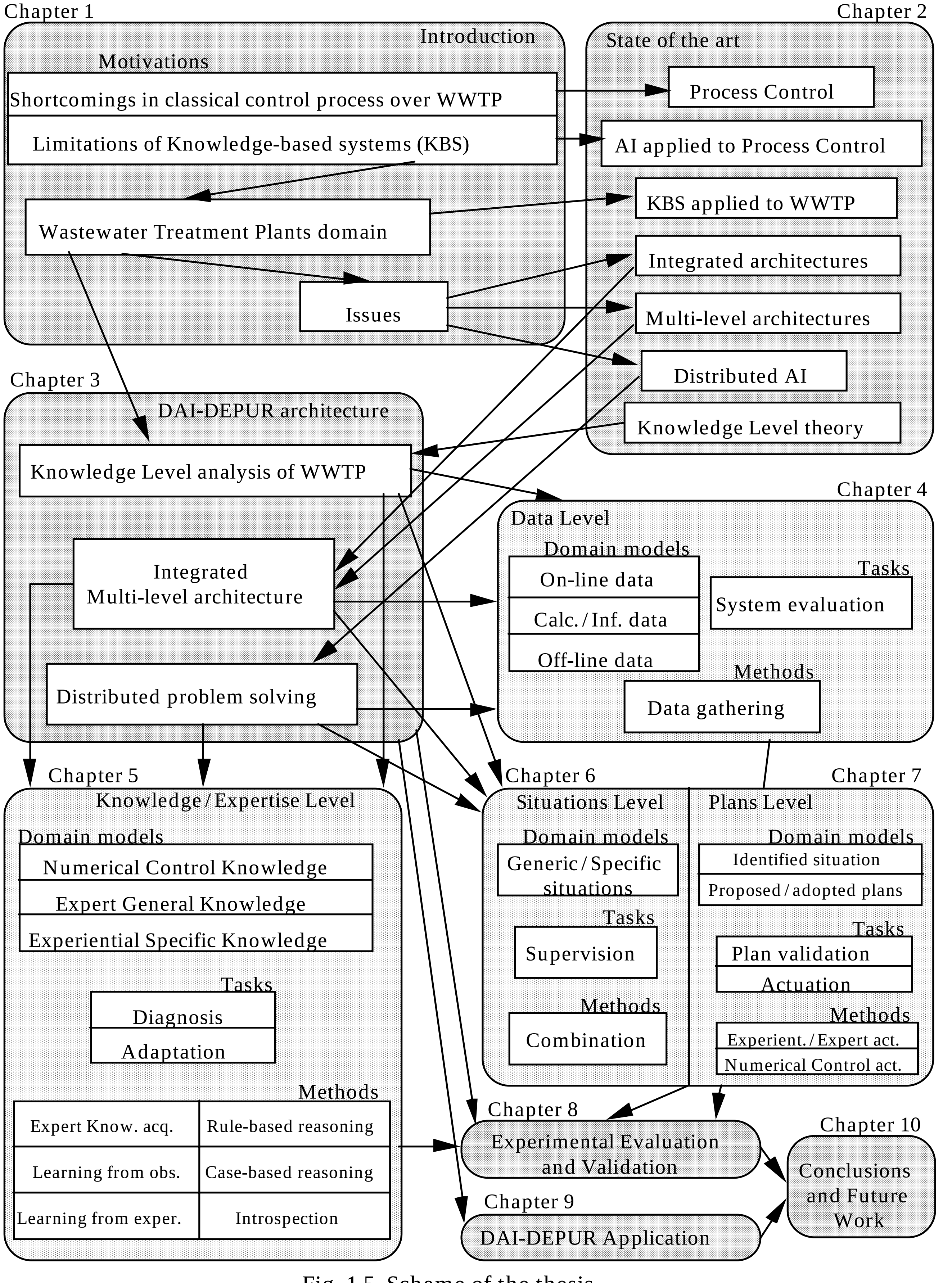


Fig. 1.5. Scheme of the thesis

Chapter 8 presents the experimental evaluation performed on the system. It has been made at two stages: single validation of the main models (numerical, expert and experiential knowledge), and whole evaluation of DAI-DEPUR at three levels, by means of simulations of real working of plants, by validation on a scale pilot plant –built -up to this proposit– and by validation on a real plant.

Chapter 9 shows in detail some aspects of the application of the system in real execution, presenting some examples of the system' running and discussing the human-computer interaction provided by the system.

Chapter 10 discusses the research work developed, presents some contributions of our work and, finally, points to future research work that can be done in this field, to improve the system.

This organization is depicted in figure 1.5.

Chapter 2

The State of the Art

2.1 Process Control

The main goal of process control is to adequate the behaviour of the system under supervision, to prefixed good system operation levels, i.e. set-points. The dynamics of the system is evaluated by means of some measures of the parameters of the system and controlled through the actuation over some variables of the system. A general chart of a control process is depicted in figure 2.1.

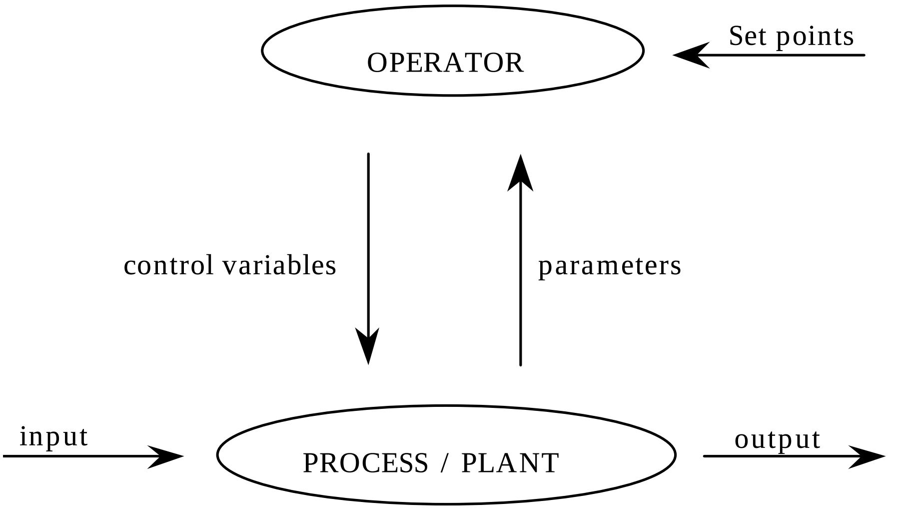


Fig 2.1. Process control chart

Automatic process control appears when a computerized agent is interleaved between the operator and the process under control. The inclusion of a computer provides the whole system with some of the following characteristics –that makes easier the operator's work–: faster supervision speed, computer numerical control algorithms, graphical interface, numerical manipulations (variable's graphics, statistical analysis, *etc.*), simulation capabilities, faster time response speed. The chart of an automatic control process is shown in figure 2.2.

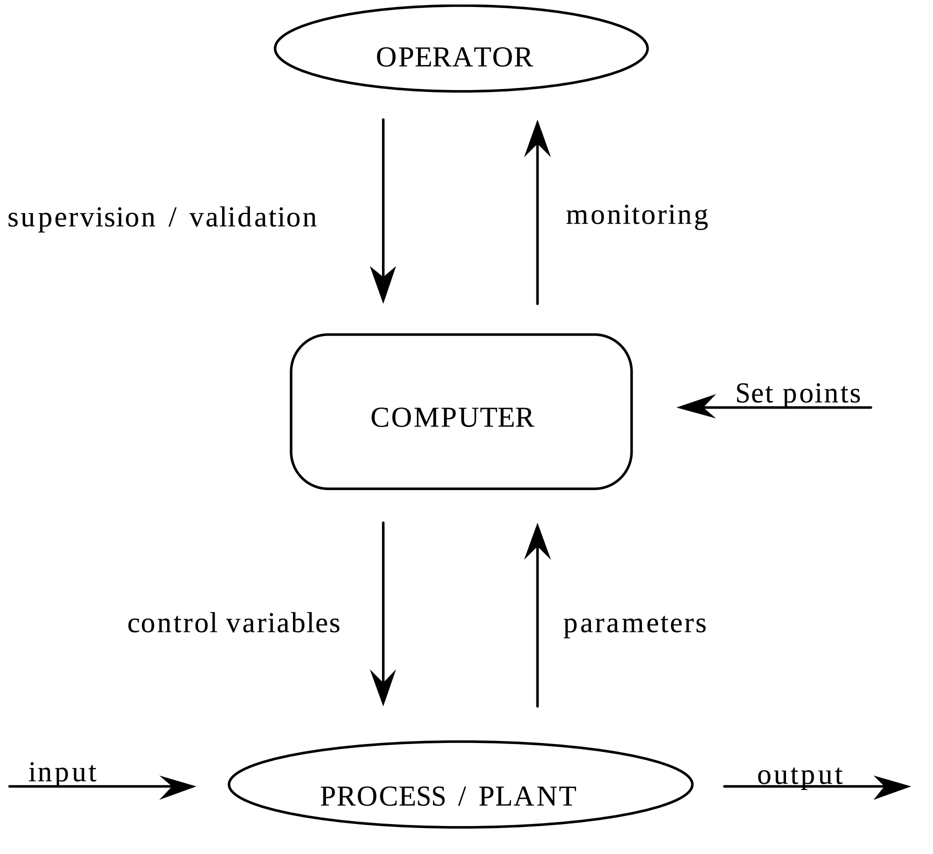


Fig 2.2. Automatic process control chart

Nowadays, most of the architectures of supervised automatic process control systems are composed by different levels of control between the process and the human operator (see figure 2.3), [Aguilar *et al.*, 1992; Aguilar, 1990]. At first level, the sensors and actuators interface with physical devices to gather some values from the process and to update some variables of it. The second level (control level) is formed by several classic control techniques (regulators, optimizers, programmable logic controllers, parameters identification, feedback control, *etc.*). The third level shows the monitoring (evolving database) of the process through visualization.

Finally, the fourth level (supervision level) provides the dialogue between the computer system and the operator.

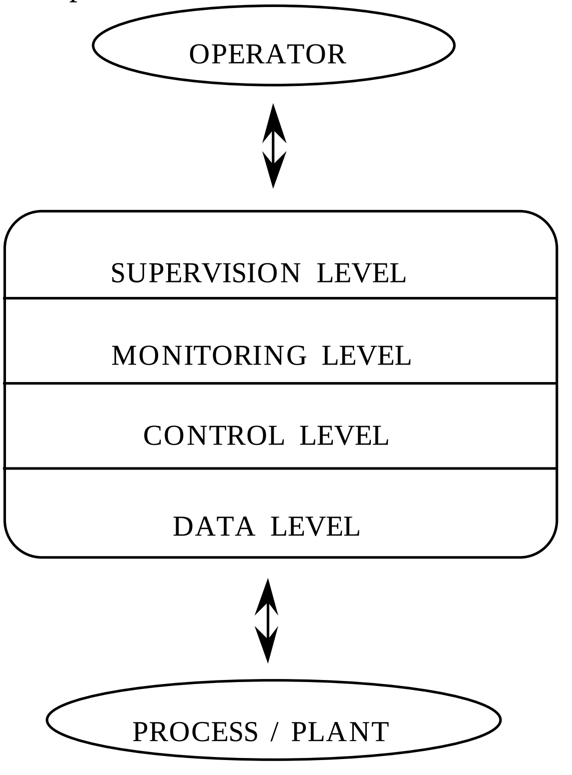


Fig 2.3. Supervised automatic process control architecture

2.1.1 Automatic Process Control applied to WWTP

During its operation, a WWTP must satisfy several requirements imposed by the generic technical, economic, and social conditions in the presence of ever-changing external influences. All these requirements dictate the need for continuous monitoring of the operation of the plant and external intervention (control) to guarantee the satisfaction of the operational objectives. This is accomplished through a rational arrangement of equipment (measuring devices, valves, controllers, computers, *etc.*) and human intervention (plant designers, plant operators, *etc.*), which together constitute the control system.
There are three general classes of needs that a control system is called on to satisfy:

- Suppressing the influence of external disturbances.
- Ensuring the stability of the process.
- Optimizing the performance of the process.

The following represent different automatic process control configurations which have been applied to WWTP:

• *Feedback control*: uses direct measurements of the controlled variables to adjust the values of the manipulated variables. The objective is to keep the controlled variables at desired levels on set-points. It is commonly used to control the DO level in aeration tanks and it has also been studied to control substrate and biomass [Marsilli-Libeli, 1982]. A feed-back controller reacts only after it has detected a deviation in the value of the output from the desired set point.

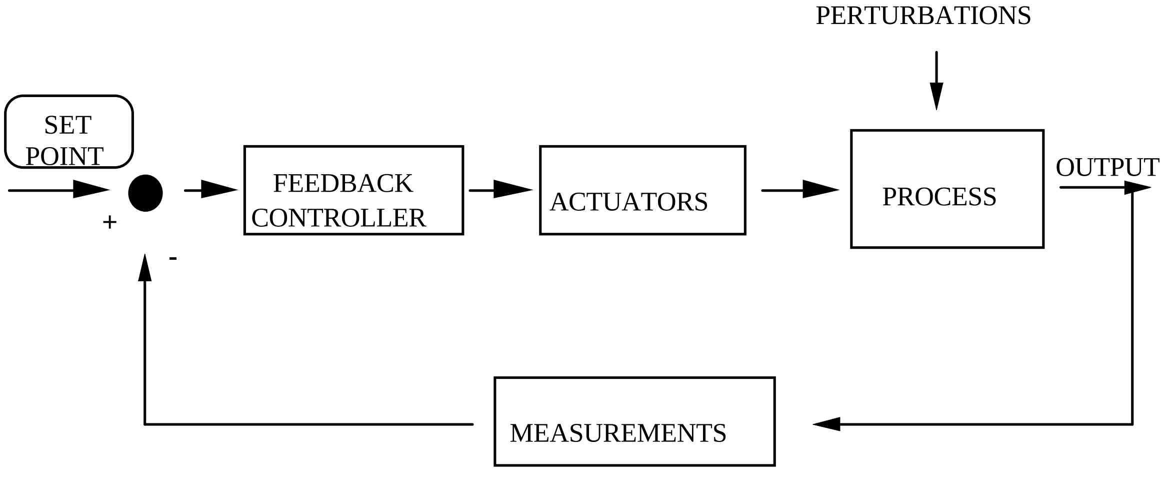


Fig. 2.4. Chart of feedback control

• *Feedforward control*: unlike the feedback systems, a feedforward control uses direct measurements of the disturbances to adjust the values of the manipulated variables. An example of feedforward control configuration was implemented in Luggage Point WWTP [Corder and Lee, 1986].

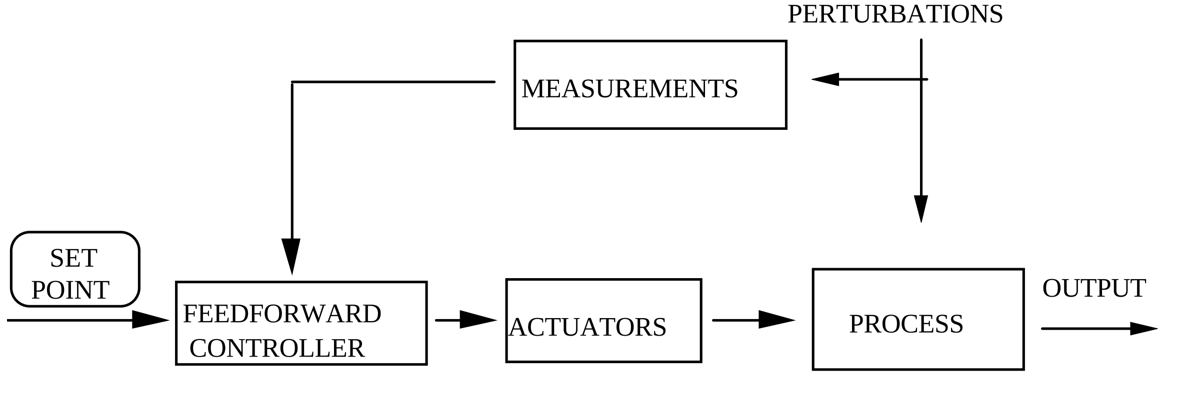


Fig. 2.5. Chart of feedforward control

• *Adaptive control*: a control system is called adaptive, which can adjust its parameters in the characteristics of the process it controls. There are two main reasons to use an adaptive controller in a WWTP. First, the process is non-linear as the desired steady-state operation of the process changes, the best values of the controller's parameters change. Second, the process is nonstationary, and thus, it changes with time. Different configurations of adaptive control have been proposed in [Dochain, 1991; Ko *et al.*, 1982].

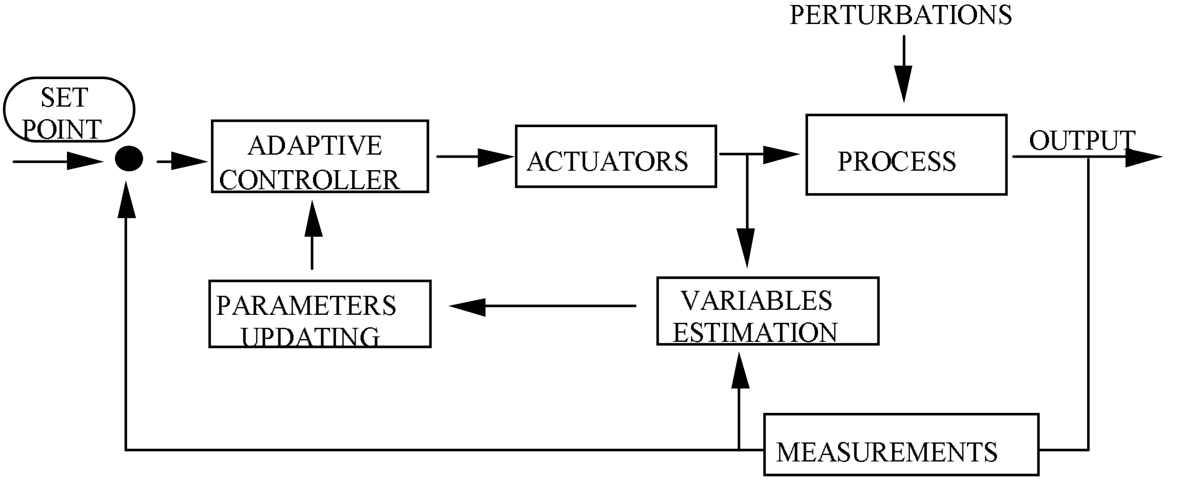


Fig. 2.6. Chart of adaptive control

• *Optimal control*, [Beck, 1986], and *predictive control*, [Moreno *et al.*, 1992; Clarke *et al.*, 1987b; Clarke *et al.*, 1987a], complete the list of Automatic Process Control configurations applied to WWTP.

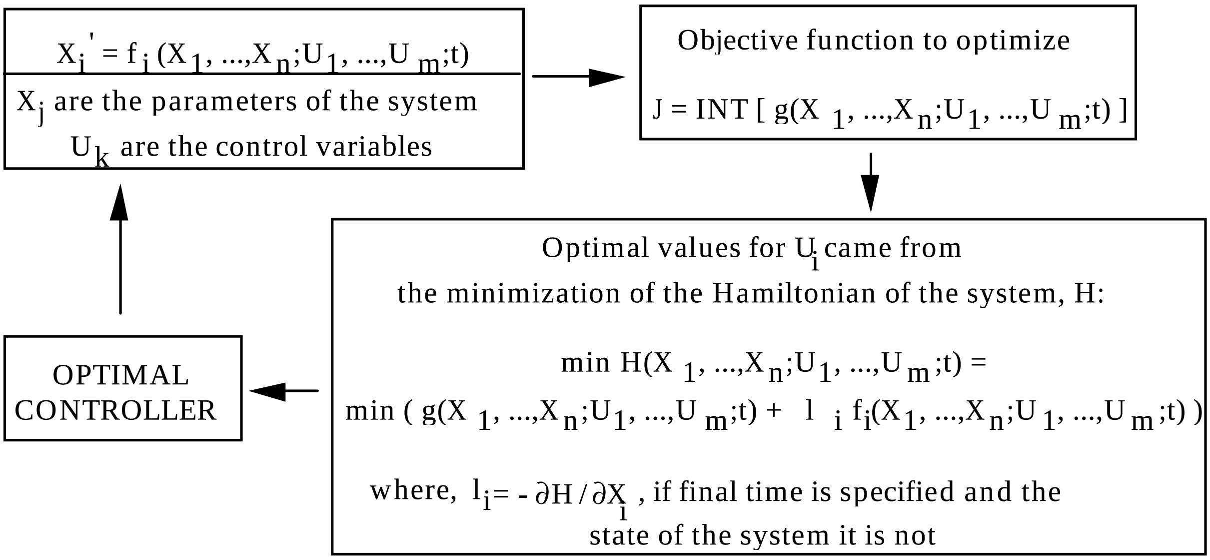


Fig. 2.7. Chart of optimal control

The complexity of the process –composed by several operational units– makes difficult the implementation of an automatic process control over the wastewater treatment plant system. There are *many factors influencing the system* and most of them cannot be controlled, as for example the water temperature, flow variations, peaks, toxic loading, *etc*. Furthermore the *domain is non-well structured*: there is a lack of understanding of the true mechanisms of the biochemical processes involved in wastewater treatment plants and the relationships among different phenomena which characterize the system are not well-enough known, although different mathematical models have been put forward to describe them.

Most information is neither numeric nor quantified; qualitative information can not be used in the context of a conventional control model, as for example microbiological information, water smell and appearance or state of the floc during sedimentation. This kind of information is essential for the operator of the plant, but is not suitable to be included in the context of a classical-numerical-control model. Another added difficulty is the *uncertainty or approximate knowledge*; the variables which describe the process are global and most of them cannot be obtained on-line.

Finally, *the system is dynamic*, it is under continuous changes that can directly modify the performance of the process. The plant is never working in steady-state conditions, and, in addition, a conventional process-control system such as feed-back control, feed-forward control, adaptive control, *etc.* cannot work when there is some mechanical fault (i.e. in the turbines, bridge of the clarifier, *etc.*) or when the information is available but incomplete or noisy.

2.1.2 Real-Time Systems

First of all, it is needed to begin defining the term *real-time system* [Stankovic, 1988; Allworth and Zobel, 1987; Bennett, 1987]. There are some definitions of real-time systems previously stated. Thus, [Young, 1982] defines a real-system as:

> any information processing activity or system which has to respond to externally-generated input stimuli within a finite and specified period.

[Burns and Wellings, 1990] define a real-time system as:

> computer applications, whose prime function is not that of information processing, but which nevertheless require information processing in order to carry out their prime function.

The Oxford Dictionary of Computing gives the following definition:

> any system in which the time at which output is produced is significant. This is usually because the input corresponds to some movement in the physical world, and the output has to relate to that same movement. The lag from input time to output time must be sufficiently small for acceptable timeliness.

[Motus, 1994] describes a real-time system as:

> a collection of co-operating dynamic systems, one of which is a computer system. The co-operation may, in some cases, be one-sided. In other cases, the dynamic systems (one of which directly interacts with the environment) truly cooperate, making thus, better use of each facilities in order to achieve their goals. Nevertheless, the computer system always take the active role by coordinating and / or monitoring the other partners so as to achieve the goals set for the real-time system. The two basic goals are to obtain a consistent set of time (and other) constraints and to ensure time deterministic behaviour of the system with the required quality of functioning.

Often in the literature, *hard real-time systems* are distinguished from *soft real-time systems*. Hard real-time systems are those where it is needed that responses occur within the specified deadline. Soft real-time systems are those where response times are important but the system will still function correctly if deadlines are occasionally missed. Another distinction can be made: *Interactive real-time systems* are different from soft real-time systems, in the sense that there are no explicit deadlines.

In real-time systems the computer is usually interfaced directly to some physical equipment and is dedicated to monitoring the operation of that equipment. The role of the computer is like an embedded system within a larger engineering system. Therefore, real-time systems have also become known as *embedded computer systems*, [Motus, 1990]. Another names used by the scientific community are *reactive systems,* [Manna and Pnueli, 1991], or *hybrid systems*, [Maler, 1992] pointing to the fact that are systems very coupled with the environment, and that have to integrate multiple tasks, domains and subsystems (domain knowledge, problem solving, reasoning, control engineering, data acquisition, on-line sensors, operator's interface, physical systems, *etc*.).

Some examples of real-time systems are:

- manufacturing process control systems
- physical devices monitoring systems (as higher/lower levels of tank control, electrovalves/pumps switch on/off control, etc.)
- airline seat reservation systems
- robotic systems
- automatic intensive patient care systems
- air traffic control systems
- remote bank accounting systems
- industrial process control systems

The main characteristics of real-time systems –that, obviously not all real-time systems will exhibit– can be enumerated as follows ([Sriram, 1992; Burns and Wellings, 1990]):

- *Numerical information*. It is necessary the manipulation of real numbers involved in the control of engineering activities (real input data, differential equations solving, simulation processes, analogue–digital converters, *etc*.).

- *Qualitative information*. This kind of information –as for example expert knowledge or experiential knowledge– is very important for controlling the physical system, although not suitable to be included within a classical numerical control model. Nevertheless, it must be supported by the real-time control system.

• *Uncertainty or approximate knowledge*. Often, some variables and parameters of the physical system are not well-known, due either to bad operating sensors or to the fact that are completely unknown and cannot be directly measured at all, and have to be estimated from other values. Furthermore, certain subjective information supplied by the experts is also uncertain.

• *Complexity*. Most of real-time systems are large real-world systems under continuous changes that can update the systems' performance. The variety of this external changes / events is very large.

• *Reliability and safety*. As real-time systems usually control crucial processes, more important is that those systems do not fail. The failure of these systems can become a great disaster, causing expensive damage to equipment, or furthermore, to environment. Thus, continued safe operation is required.

• *Interactive systems*. As real-time systems control real-world systems, they are intended to interact with physical systems as for example monitoring sensors and controlling actuators.

• *Real-time handling*. Meeting time response within predefined intervals of time, is a major requirement of real-time systems. Handling of asynchronous events or exceptions is also needed in a dynamic and unforeseen domain.

• *Efficient implementation*. High performance is more important in real-time systems than in other ones, due to hard constraints (time or others).

• *Concurrent system components*. There are a set of coexisting elements or processes in a real-time system. All this subsystems act at the same time, *i.e.*, in parallel or concurrent execution. Although with high speed computers, this parallelism can be implemented in a sequential way, there are systems in which this may not be the case, due to hard time constraints or to really distributed sites processing.

2.1.2.1 Are WWTP Real-Time Systems ?

In addition to the above features, wastewater treatment plants are *ill-structured domains*, where the relationships among the different elements of the process are not well enough known and the dynamics of biochemical processes involved is not well understood, as previously described.

Bearing in mind these features and those ones from WWTP described in chapter 1, it can be argued that wastewater treatment plants management systems are real-time systems. They are not strictly hard real-time systems so that in wastewater treatment plants, there is no hard time constraints. Luckily, the operating dynamics of wastewater treatment plants is slower than input variability in both inflow quantity and quality and consequently, can stabilize these continuous changes and deviations into the system. Therefore, it is not crucial to respond to these events within brief periods of time. But, which is really important is to get continuously low values from output COD and output BOD that is lower than the environmental law's bound, i.e. 20 mg/L. If these low values are not achieved, then great damage can be caused to the environment and to human beings. So, there are *hard output water quality constraints or good plant operation constraints* (see Appendix B). In this sense, it can be said that wastewater treatment plants are *very hard real-time systems*.

2.2 Artificial Intelligence applied to Process Control and Supervision

Artificial Intelligence can cooperate with classical automatic control systems and classical real-time systems to get a *real-time intelligent control system*, able to cope with either hard real-time/other constraints or with ill-structured domains or with some features not suitable to be integrated in classical control systems as learning, reasoning, modelling of expertise, qualitative information, uncertainty, *etc.*.

What is *Artificial Intelligence*? It is not easy to define this complex and widely used term. [Rich and Knight, 1991] define it as follows:

> Artificial Intelligence (AI) is the study of how to make computers do things which, at the moment, people do better. This definition is, of course, somewhat ephemeral because of its reference to the current state of computer science. And it fails to include some areas of potentially very large impact, namely problems that cannot now be solved well by either computers or people. But it provides a good outline of what constitutes artificial intelligence, and it avoids the philosophical issues that dominate attempts to define the meaning of either *artificial* or *intelligence*.

[Charniak and McDermott, 1985] define it as:

> Artificial Intelligence is the study of mental faculties through the use of computational models. The occurrence of the word "intelligence" in the name of this field is misleading. When talking about what other people do, we tend to reserve the word for mental feats of unusual creativity or cleverness. As a consequence, it sounds as if *artificial intelligence* were a technique for producing an abundance of clever insights. In fact, the most interesting for AI arise in attempts to duplicate the mental faculties of "ordinary" people.

[VUB AI Lab, 1993; Steels, 1985] describe artificial intelligence as:

> Artificial Intelligence is a scientific research field concerned with intelligent behaviour. AI researchers want to develop precise explicit models of the structures and processes that give rise to intelligent behaviour, and apply their insights to the constructions of artefacts that are useful to the people. These artefacts range from knowledge systems, which support problem solving activities such as scheduling, to autonomous mobile robots, which help in jobs that may be hazardous or tedious for humans. What makes Artificial Intelligence unique, compared to other sciences that also study intelligence like psychology or neurophysiology, is the strong emphasis on the construction of artificial systems as a way to test theories.

Perhaps a wider definition could integrate all points of view: *artificial intelligence* is the study of the possible or existing mechanisms –in human or other beings– providing such behaviour in them, that can be considered as *intelligence*, and the emulation of these mechanisms –usually called cognitive tasks– in a computer through the computer's programming.

Major assumptions in classical symbolic paradigm of Artificial Intelligence rely on these facts:

- What the brain does, may be thought of –at some level– as a kind of computation.
- The *physical symbol systems hypothesis* [Newell and Simon, 1976]. A physical symbol system has the necessary and sufficient means for general intelligent action.
- The *knowledge level hypothesis* [Newell, 1982]. There exists a distinct computer systems level, lying immediately above the symbol level, which is characterized by knowledge as the medium and the principle of rationality as the law of behaviour.
- The *principle of rationality* [Newell, 1982]. If an agent has knowledge that one of its actions will lead to one of its goals, the agent will select that action.

These *cognitive tasks* –starting from stimuli perception tasks and ending with response action tasks, passing over some internal cognitive tasks– are: vision, natural language, knowledge acquisition, knowledge representation, reasoning, search, planning, explaining, learning, motion (robotics) and speech. Therefore, each one of these tasks has its own specific problems and has developed its own methodologies.

A computational system could be considered as an intelligent system if exhibits some, or better, all of the following human intelligence characteristics listed by [Newell, 1990]:

- Behave flexibly as a function of the environment
- Exhibit adaptive (rational, goal-oriented) behaviour
- Operate in real-time
- Operate in a rich, complex, detailed environment (perceive an immense amount of changing detail, use vast amounts of knowledge, control a motor system of many degrees of freedom)
- Use symbols and abstractions
- Use language, both natural and artificial
- Learn from the environment and from experience
- Acquire capabilities through development
- Operate autonomously, but within a social community
- Be self-aware and have sense of self
- Be realizable as a neural system
- Be constructible by an embryological growth process
- Arise through evolution

There are a great variety of Artificial Intelligence paradigms or approaches that have been proposed –since the beginning of artificial intelligence research– to model human intelligence, as: Logic paradigm, Heuristic search and Planning paradigm, Knowledge-Based paradigm, Model-Based paradigm, Experience-Based paradigm and Connexionist or Subsymbolic paradigm.

2.2.1 Intelligent Control

In the supervision level, see figure 2.3, of a supervised automatic process control systems, is where some artificial intelligence techniques can be integrated to improve the performance of an automatic control systems [Rao, 1992; Stock, 1989; Laffey *et al.*, 1988]: controlling abnormal situations, uniforming the solutions given by different operators, improving the execution speed of the supervision cycle, providing computer aided assistance to operators, *etc*. There some features arguing for this approach usually named as*intelligent advanced control systems*:

- Uncertain or approximate reasoning
- Non-analytical representation
- Data-driven process
- Qualitative information
- Auto-modification ability
- Non-determinism
- Justification and explanation facilities
- Search algorithms
- Learning
- Using experience, judgements and human expertise

From an historical point of view, one can distinguish some features in the developing of intelligent advanced control systems:

- The work of K.S. Fu about regulators [Meystel, 1985a; Fu, 1971].
- Theoretical developments by university research groups, regarding to the integration of artificial intelligence, control theory and hierarchical systems [Tzafestas and Ligeza, 1989; Åström *et al.*, 1986; Meystel, 1985b; Saridis, 1985].
- Expert systems application to real world systems [Bonissone, 1993; Brajnik, 1989; Bernard, 1988; Shirley, 1987; Intelllicorp, 1986].
- Appearing of specific tools for supporting reliable developing of intelligent control systems such as RTworks from Talarian Corporation, G2 [Gensym, 1992; Gensym, 1990], PAMELA-C [Barachini and Theuretzbacher, 1988], HEXSCON [Wright, 1986], PICON [Moore *et al.*, 1984], etc.
- Some research projects that try to improve intelligent control systems by providing reliable real-time operation and wider applicability [Cavanna *et al.*, 1989; Voss, 1988].

Main usually applied intelligent control techniques are [Sànchez *et al.*, 1995c; Kuhn, 1971]:

- Knowledge-Based control. It will be described in next subsection.
- Fuzzy control. There are several works about application of fuzzy sets theory ([Zadeh, 1983; Zadeh, 1979]) to control systems such as [Bonissone, 1994; Piskunov, 1992; Bouslama, 1992; Czoagala and Rawlik, 1989; Tagaki, 1985; Sugeno, 1985].
- Model-Based control such as the work of [Ramparany, 1994; Rich and Venkatasubramanian, 1987].
- Neural network control. Some recent works are [Hunt *et al.*, 1992; Kraft *et al.*, 1992; Kosko, 1992; Capodaglio *et al.*, 1991; Narendra and Parthasarathy, 1990].
- Genetic control, such as the work of [Karr, 1991; Karr *et al.*, 1989].

But, most of them have been only used for either improving numerical algorithms or diagnosis stage. Bearing in mind some previously outlined features of wastewater treatment plants, and *the issue of management or supervision of WWTP*, one can realize that the most suitable approaches for whole WWTP supervision are both Knowledge-Based and Experience-Based ones. Most of the applications developed in the field are based in the Knowledge-Based paradigm. For this reason, a survey on some of these applications will be made in next section.

2.2.2 Knowledge-Based Systems applied to WWTP

Specially related to wastewater treatment area [Alleman *et al.*, 1992; Patry and Chapman, 1989; Kodukula, 1988], most of Knowledge-Based applications have been developed as off-line consultations –in an increasing chronological order– among other tasks for:

a) Design: [Krovvidy *et al.*, 1991; Krovvidy and Wee, 1993];
b) Diagnosis and Decision-aid: [Beck *et al.*, 1978; Flanagan, 1979; Flanagan, 1980; Maeda, 1985; Berthuex *et al.*, 1987; Gall and Patry, 1989; Lapointe *et al.*, 1989; Maeda, 1989; Beck *et al.*, 1990; Krichten *et al.*, 1991; Sànchez, 1991; Belanche *et al.*, 1992b; Serra *et al.*, 1994]; and
c) Process optimization: [Huang *et al.*, 1991].

2.2.2.1 Design Systems

[Krovvidy *et al.*, 1991] propose a two phase approach to the design of wastewater treatment systems. In the first phase, called *analysis phase*, they developed a learning system to generate knowledge rules from a treatability data base. Also, they have developed a grammar-based knowledge representation scheme to be able to generate rules for different expert systems shells. In the second phase, called synthesis phase, developed two different methodologies: a heuristic search approach and a neural network approach to generate treatment training sequences with minimum cost. The heuristic search function is developed based on the removal capabilities of the treatment processes. The neural network model is developed by formulating the synthesis phase as an optimization problem and an energy equation is derived for Hopfield network to generate the treatment training sequences. Both approaches are compared for the optimality of the solution and the processing time required.

[Krovvidy and Wee, 1993], formulate, the design of wastewater treatment systems as a heuristic search problem. Heuristic search is one of the most widely used techniques for obtaining optimal solutions to many real-world problems. In that paper, they identify some necessary properties of the heuristic search problems to be solved in the Case-based reasoning paradigm. Case-based reasoning (CBR) is one of the emerging paradigms for designing intelligent systems. Preliminary studies indicate that the area is ripe for theoretical advances and innovative applications. They designed a CBR system based on that properties and performed several experiments with the wastewater treatment problem. A comparison between the CBR system and the A^* search algorithm performance is provided in their work.

2.2.2.2 Diagnosis and Decision-Aided Systems

[Beck *et al.*, 1990; Beck *et al.*, 1978] were among the first to make use of expert system-type rules for wastewater treatment plant operation and control. Although the authors do not claim to be using expert systems, they employ the crucial aspect of expert systems, human expertise or expert knowledge. Twenty heuristic control rules were formulated following discussions and claims of two of the authors, one being a treatment plant operator. Fuzzy logic was used to provide a qualitative interpretation of the quantitative data.

[Flanagan, 1979; Flanagan, 1980] developed a system for the activated sludge process, that use fuzzy reasoning but employed an operating strategy proposed by [Olsson and Andrews, 1977], that relies on a profile of dissolved oxygen (DO) concentration along the length of the reactor as an indicator of biological activity and reactor loading. The DO profile strategy is based on a mechanistic understanding of the process. Thus, in contrast to the heuristic knowledge represented in other systems previously developed, his system made use of compiled knowledge.

[Berthuex *et al.*, 1987] based on previous work by Beck *et al.*, extended that work by integrating the expert system to a database to provide plant operators with a more powerful software package. An outstanding feature of that system was that the expert system could be customized to a particular treatment process. It was supplied a wide variety of rules for different treatment processes. The knowledge base was constructed by specifying the configuration of the plant.

[Maeda, 1985; Maeda, 1989] presents a Knowledge-Based decision support system for wastewater treatment plants. The system get benefits from the operator's qualitative and experienced judgement in supervisory tasks such as set point schedulling, plant diagnosis and maintenance. The architecture of the system is composed by two parts: one is an adaptive production system about 100 production rules, and the second is a multimodal user interface including video graphics, voice announcement, touch panel and mouse. The performance of the system through typical examples of operational guidance is demonstrated. Maeda concludes that the proposed system would be flexible and pragmatic: the human operator keeps a central role in maintaining the high reliability of future biological systems and the use of the system will increase the operator's cognitive capabilities.

[Gall and Patry, 1989] describe the development of a Knowledge-Based System for the diagnosis of an activated sludge WWTP. The knowledge base was developed using two basic sources of information: literature review of wastewater treatment plant operation, and site visits and interviews with experienced plant operators. The system was developed using TI Personal Consultant Plus^{TM} and tested under actual plant operating conditions, the knowledge base consists of 169 rules based on readily available information, that is deeply explained in the paper. In general, feedback obtained from the operators confirmed the potential benefits of expert-assisted operation of wastewater treatment plants. They also stated that the operational benefits of a Knowledge-Based System for the activated sludge diagnosis depends largely on the continuing contributions from plant operators. They postulate that the knowledge base should not be viewed as a static piece of software but should be updated on a regular basis to reflect the cumulative experience of the operators as well as changes and/or adjustments made to the different unit processes.

[Lapointe *et al.*, 1989] described a Knowledge-Based System for the upflow anaerobic sludge blanket process, which organizes behavioural, functional and structural knowledge in an object-based manner. In that scheme, frames are defined that can have methods as attributes. These methods may be either rules or may implement a quantitative model. Therefore, there is a deep model-based knowledge and a shallow heuristic-based knowledge. A generate-and-test reasoning strategy is used, and explanation facilities are also provided. The authors argue that their system achieves the goals of process generality, fault diversity, reasoning transparency, reliability and graceful degradation.

[Sànchez, 1991; Belanche *et al.*, 1992b; I. R.-Roda, 1994; Serra *et al.*, 1994] present the development of a prototype of fuzzy expert system useful for the diagnosis and management of wastewater treatment plants that has evolved to the current architecture, now called DAI-DEPUR. First, it is described the complexity of the system that is being modelled in order to outline its own difficulties. The development of the qualitative expert system in the shell MILORD [Sierra, 1989] is presented: attribute selection and knowledge acquisition. A new methodology used for automatic knowledge acquisition is introduced. It is used for building-up the knowledge base. Some details of the architecture, user interface and implementation are provided. The prototype diagnostics were validated against the human operators judgements, yielding very good results.

2.2.2.3 Control Systems

More recently, Knowledge-Based techniques begin to be applied for on-line process control and supervision, in the field of WWTP, as for instance:

[Serra *et al.*, 1995; Serra *et al.*, 1993] present a real-time expert system to control wastewater treatment plants. The software has been developed in the G2 shell. The system is composed by an interface that let on-line acquisition of plant data using G2 standard interface, a predictive control algorithm for dissolved oxygen (DO), a graphical interface for the operator and the expert knowledge. The dissolved oxygen control is performed using a non-linear predictive control algorithm, that has been developed to satisfy quality constraints while reducing energy demands. The algorithm uses data obtained from the plant by hardware sensors, and software which recursively estimates the oxygen uptake rate (OUR). All these elements are integrated in a knowledge base that includes a set of diagnosis, detection, prediction and operation rules, making the system capable of handling a wide number of usual situations where predictive control can be useful, and unusual situations where quantitative and qualitative information must be considered. Also this research has evolved to the current architecture, DAI-DEPUR.

2.3 Knowledge-Level Theory

The knowledge-level notion was introduced by [Newell, 1982]. He stated that in a cognitive system, the *knowledge* is to be characterized entirely functionally, in terms of what it does, and not structurally, in terms of physical objects with particular properties and relations. So, the knowledge level is an abstract level of description lying above the symbol or programming level, which aims to better knowledge engineering, knowledge acquisition, understanding and analysis of Knowledge-Based Systems disregarding the concrete computational aspects.

In the past decade, several ideas have emerged that go in the direction of the knowledge level analysis of knowledge/expertise: the concept of an inference structure [Van de Velde, 1987a; Clancey, 1985b], the distinction between deep and surface knowledge [Steels, 1985; Hart, 1984], the decomposition of expertise into problem solving methods and domain knowledge filling the roles of these methods [McDermott, 1988], and the notion of generic task [Chandrasekaran, 1983]. All these ideas have led to focus on the knowledge itself, rather than on the information processing (implementation) details of a system. All these approaches have been synthetized in the componential framework of expertise [Steels, 1990]. In next subsections we briefly review all these knowledge-level approaches.

2.3.1 Generic Tasks

One line of research focuses on *task* features and, thus, directly addresses the problem of developing an engineering methodology to build knowledge-based expert systems based on a task analysis. The analysis of expertise in terms of tasks (and, particularly, the ordering of the tasks, that is, the control structure imposed on the task structure) used to be a completely domain-dependent matter, enforcing the view that Knowledge-Based System development does not show generalizations across domains of expertise and, therefore, is doomed to start new, in an *ad hoc* fashion, for every new application being tackled. However, several researchers have observed that tasks fall into major classes. These tasks are called *generic tasks* [Chandrasekaran, 1983]. In specific fields of expertise, tasks are instances of these generic tasks. Typical generic tasks are classification, interpretation, diagnosis and construction (including planning and design). The idea of generic tasks is not that interesting in itself until we realize that the way in which generic tasks are executed shows many similarities across application domains: in the diagnosis of circuits, cars, power plants, or diseases, significant elements are in common, specifically, the same problem-solving methods and the same type of domain models. Another interesting point is the division of a task into simpler *subtasks* to accomplish the goal of the task.

2.3.2 Inference Structures

An *inference structure* describes the pattern of inferences found in a particular Knowledge-Based System. *Heuristic classification* is the most widely studied class of inference structure [Van de Velde, 1987a; Clancey 1985b]. Heuristic classification assumes three major inference types: those making *abstraction* of the data, those *matching* the data with an abstract solution class, and those *refining* this solution to the actual solution. How theses inferences are made –for example, by one rule or many or by other kinds of inference mechanisms– is not an issue, although the analysis includes a characterization of the kind of relation that is used to perform the inference. Data abstraction makes use of qualitative, definitional and generalization relations. Heuristic match is based on a causal relation. Refinement is based on a subtype relation.

Inference structures show that there is a lot of structure underlying the rules in expert systems, including many hidden assumptions. Also, they describe relationships between rules that go beyond the syntactically based rule generalizations, and show the similarities between expert systems constructed for apparently widely diverse domains and tasks.

2.3.3 Deep versus Surface Knowledge

Another type of analysis was proposed based on a distinction between deep and surface knowledge [Steels, 1985; Hart, 1984]. This distinction focused not on the pattern of inference but on the *domain models* underlying expertise. *Deep knowledge* makes explicit the models of the domain and the inference calculus that operates over these models as for example causal, associational, functional models, *etc.*. A typical example of a domain model for diagnosis is a causal model linking properties of components through cause-effect relations. An inference calculus operating over this model could take the form of a set of axioms that prescribe valid inferences over the causal network, for example, inferences showing that a certain cause possibly explains a specific set of symptoms. *Surface knowledge* contains selected portions of the deep knowledge, in particular, those portions that are relevant for the class of problems that is likely to be encountered. It also contains additional heuristics and optimizations, for example, shortcuts in the search space or decisions based on the most probable situation. Traditional Knowledge-Based Systems only contain surface knowledge, while the so-called second generation (deep) Knowledge-Based Systems usually have both of them.

2.3.4 Problem-Solving Methods

[McDermott, 1988] and his colleagues developed a series of knowledge-acquisition tools that emphasized the problem solving method and not the inference pattern or the domain model, as the central key in understanding and building an application. A *problem solving method* is a knowledge-level characterization of how a problem might be resolved. For example, diagnosis might be done using the cover-and-differentiate method: first, find possible explanations covering most symptoms, and then, differentiate between the remaining explanations. Construction also might be done using the propose-and-revise method: first, propose a partial solution, then revise this solution by resolving violated constraints. Each problem solving method contains certain roles that need to be filled by domain models. For example, the cover-and-differentiate method requires knowing relationships between explanations and the symptoms they cover and knowing additional observations or tests that will further differentiate. The propose-and-revise method implies two roles: one to be filled by knowledge that is proposing solutions, another to be filled by knowledge that revises these solutions.

2.3.5 KADS

The KADS project, funded by the EEC, concerns both a methodology for knowledge modelling and a framework for knowledge acquisition, design and implementation of knowledge systems [Wielinga *et al.*, 1992]. Four representational layers are distinguished: *domain* layer, *inference* layer, *task* layer, and *strategy* layer. In the domain layer, the domain objects and their relations are modelled. In the inference layer, there is an explicit description of the roles that an object can play in the problem solving process. A taxonomy of generic elementary problem-solving roles is identified. In the task layer, a series of operators corresponding to the roles described in the previous layers is given. This set of operators enables the achievement of the model goals. In the strategy layer, different combinations of operators may be obtained from the representation of meta-plans.

2.3.6 Componential Framework of Expertise

L. Steels proposed the combination of all above explained ideas [Steels, 1990] for knowledge-level analysis of expertise, in what he called *components of expertise*. He suggested that a problem domain can usually be described from three basic perspectives: tasks, models and methods (see figure 2.8)

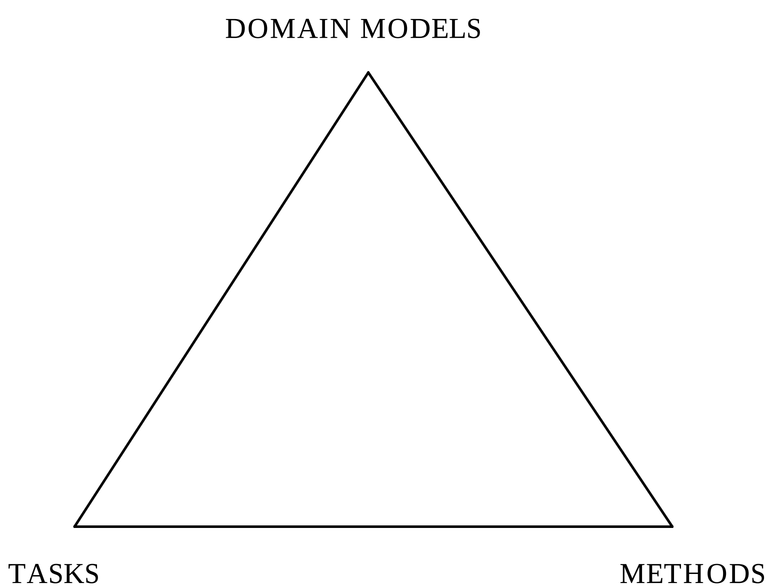


Fig. 2.8. Components of expertise

The *task analysis* is concerned with a conglomerate of mutually dependent tasks, and with internal structure of tasks: they can be decomposed into subtasks with input-output relations between them. Each task and subtask is analyzed from a conceptual and a pragmatic point of view. The perspective of deep Knowledge-Based Systems has made researchers aware that problem solving centers around the idea of modelling. The problem solver constructs one or more *models* of the problem solving situation (called the *case model* by Steels) and uses various more abstract *domain models* to expand the case model by inference or data gathering. Both kind of models integrate, what it can be named as the *domain theory*, in which it can be found structural, topological, behavioural, fault, repair models, *etc.* The *problem-solving methods* apply domain knowledge to accomplish an intended task. In general, they perform two functions: divide a task into a number of subtasks or directly solve a subtask. In either case, they can consult domain models; create or change intermediary problem-solving structures; perform actions to gather more data, for example, by querying the user or performing a measurement; and expand the case model by adding or changing facts.

Recently, this methodology has evolved into the *componential framework* (COMMET) and the KREST workbench [Geldof *et al.*, 1993], whose goal is to support the design of knowledge systems on the basis of the components of expertise methodology. This ongoing research aims to support the development of an application at three levels: the knowledge level, the execution level that implements the knowledge-level components, and the code level that corresponds to a set of files containing code for the previous computational objects. The development of a small KREST application in the domain of agriculture control is reported in [Polianova, 1994]. The COMMET research is intended to experiment with reusability. It aims at making the construction of knowledge systems accessible to non-programmers. Reusability is understood as the insertion of parts of an old application into a new one. The workbench is able to seek a chunk of an old application that would be capable of carrying out the current task, adapting the old one to the new use.

2.4 Integrated Architectures

Artificial Intelligence research has produced models of the different components or cognitive tasks thought to be required for intelligence such as vision, knowledge acquisition, knowledge representation, natural language, reasoning, search, planning, learning, neural computation, genetic programming, motion (robotics) and speech. Isolating these components is vital to identify the important issues in each area, understanding those issues, and proposing solutions for them. But none of these subfields have been solved to a sufficient degree of satisfaction [Gil, 1991].

There are many systems that can be called intelligent in the sense that they produce interesting behaviour [Lenat and Brown 1983; Lenat, 1982]. As H. Simon observed [Simon, 1969] complex behaviour can arise from a simple mechanism within a complex environment. This seems to be the case when AI programs produce interesting behaviour. Other systems can be called intelligent because they give intelligent results, such as the expert systems. However, neither interesting behaviour nor intelligent results completely fulfils the requirements for truly intelligent systems. They cope with a lack of flexibility or adaptability, the knowledge acquisition bottleneck, a lack of graceful degradation (brittleness), a lack of learning from experience and the environment, a low reusability, *etc.*, that become harder when faced against real world domains. Truly intelligent systems are those exhibiting as much as possible the characteristics of human intelligence, listed by A. Newell [Newell, 1990] (see section 2.2).

In recent years, there has been an increasing interest in the integration of different aspects of intelligence in systems known as *integrated architectures*, which are capable of producing general intelligent behaviour [Van Lehn 1990; Newell 1990]. The integration of different features of intelligence will give AI researchers a better insight into how to build systems that produce intelligent behaviour.

Integrated architectures are distinguished from other systems by their intelligent behaviour. Such behaviour can be described [Newell, 1990] in terms of a system possessing knowledge and behaving in light of it. Integrated architectures should be able to handle various tasks of different nature and complexity. Domains of application may range from automatic algorithm design to planetary exploration. Many tasks require vast amounts of knowledge, and research on very large knowledge bases is just beginning [Lenat and Guha, 1990]. A useful *knowledge representation* system is not just one that can contain different kinds of information. In addition, the knowledge must be both easily accessible having good indexing and sensible retrieval time, and useful for efficient inference. Tasks may also vary in complexity. Different inference capabilities (*problem solving techniques*) may be required and, most importantly, inference must be influenced by efficiency constraints. It is important that such architectures are *efficient*. Fast mechanisms for inference and access to existing knowledge is only the beginning. It is also required that the efficiency of these systems improve through experience. This process of skill acquisition would help to avoid recomputing inferences similar to ones already made. This type of *learning* ability is desirable in an intelligent system.

Autonomy is a crucial issue for integrated architectures. It is desired to build systems that require a minimum of human assistance and maintenance. This requires the systems to have interfaces for collecting necessary data from the environment and for performing whatever actions the system decides to execute. The interaction with the environment for some tasks can be done via *natural language*, and this is desirable if the system has to communicate with humans. Other applications require that the architecture has some physical interface that allows both *perception* and execution of actions (*actuation*) in the environment. Another feature is the ability to detect failures and to recover from them. Failures can be of different complexity, requiring more or less complicated recoveries. Meta-level or *introspective reasoning* is also important. If there is an impasse in the system (conflictive goals, unexpected events, *etc.*), then some mechanisms to address that problems must be integrated in these architectures, usually by means of a reflexive or introspective reasoning.

Therefore, the combination of several components, cognitive tasks, and approaches into the same integrated architecture, make them more powerful than single paradigms used in the classical AI systems. Moreover, in the Machine Learning community there is a similar insight of what has to be the future of machine learning: *integrated learning architectures* [Plaza *et al.*, 1993], that are defined as systems able to learn and to perform at least one problem solving task, and either learning and problem solving must be flexibly integrated in a single control structure, or learning and problem solving must flexibly use the same knowledge structures.

The integrated architectures that have been proposed in the AI literature are very different. Some of them have single learning mechanisms whereas others have many learning mechanisms; the execution control can be centralized or decentralized; the decomposition of the architectures in layers or levels can be horizontal or vertical; the learning process can be automatic or directed. They could be divided into three major groups as suggested in [Plaza *et al.*, 1993]:

- *Cognitive architectures* such as SOAR [Newell, 1990; Laird *et al.*, 1987], TCA [Simmons, 1990], PI [Holland *et al.*, 1986], ACT* [Anderson, 1983], *etc.*;

- *Architectures integrating learning and problem solving* such as THEO [Mitchell *et al.*, 1990], PRODIGY [Carbonell *et al.*, 1990], ICARUS [Allen and Langley, 1990], Case-based reasoning systems like JULIA [Hinrichs, 1992] or PROTOS [Bareiss, 1989], *etc.*;

- *Multistrategy learning* [Michalsky and Tecuci, 1991].

In next subsections we briefly review most outstanding features of some of these architectures, based on the analysis performed in [Gil, 1991].

2.4.1 SOAR

The SOAR architecture [Newell, 1990; Laird *et al.*, 1987] is intended as a psychological theory of human cognition. In SOAR, all goal-oriented behaviour is formulated as search in problem spaces. Search proceeds in steps from an initial state through state-transforming operators to reach a desired state that achieves the goal. Each goal defines a problem solving context together with the definition of roles for a problem space, a state and an operator. Contexts are kept in SOAR's working memory, and problem solving is focused on selecting problem spaces, states and operators for the roles in the context. Knowledge is stored as productions in the long-term memory.

A step in problem solving is determined by a 2-phase decision cycle. In the elaboration phase the rules of the production memory are matched against working memory. The resulting instantiations of the productions are executed in parallel, adding information to the working memory that includes preferences for problem spaces, states or operators. The elaboration phase ends when no more productions can fire. The decision procedure, which is the second phase of the decision cycle, is then used to interpret the preferences in the working memory. If a role can be uniquely filled, a decision can be made, and the next decision cycle is entered. If the preferences are incomplete or inconsistent, then the decision procedure will cause an *impasse*. An impasse indicates the long-term memory did not contain unequivocal knowledge of how to proceed in this problem space. The system now automatically engages in the task of resolving the impasse, specified by a subgoal and an associated problem solving context. Given this subgoal, SOAR will bring all its knowledge to bear to try to solve it and overcome the impasse. Impasses can occur within impasses, which produces a subgoal hierarchy whose top goal is the original task goal.

Chunking [Laird *et al.*, 1986], a phenomenon that emerged from studies of human behaviour, is SOAR's single learning mechanism. Chunks are acquired each time an impasse is solved, and they summarize the process to solve the subgoal. SOAR backtraces through the productions that generate the subgoal, and the productions that generated its conditions until the elements prior to the subgoal are identified. These conditions together with the subgoal of the impasse are used to build the productions that form new chunks. Thus, the next time problem solving reaches the point at which the impasse occurred, the chunk will fire, placing its results directly with working memory and avoiding the search in the spaces below.

The SOAR architecture is based on a principle of uniformity: all tasks are implemented as search in problem spaces, all impasses result in learning, and all learning is done by chunking. The approach has been applied to a variety of tasks that include algorithm design, computer configuration, robot-arm control, and language comprehension.

2.4.2 Task Control Architecture

The Task Control Architecture (TCA) [Simmons, 1990] is being designed as a domain independent system for mobile robot control, but it can potentially be used for a wide variety of robots and tasks. The architecture is composed of several modules that communicate through a centralized control. The modules are distributed processes that enable the concurrent execution of activities such as planning, sensing, monitoring, and plan execution. This allows the system to detect important changes in the environment, to respond to them in an adequate amount of time, to detect failures and recover from them autonomously, and to consider which goals to attend to by schedulling appropriate tasks to achieve them. Tasks are specified as task trees, which are goal hierarchies combined with temporal constraints that facilitate the combination of planning and execution.

TCA is composed of several layers with different functionality. The communication layer supports the transmission of messages among processes through the centralized module. This module decides which modules handle particular messages. A behaviour layer is used to specify primitive behaviours through messages that query the environment, specify goals, and change the internal state. The task management layer analyzes these messages and constructs the hierarchical task trees. Other layers include mechanisms for monitoring, resource management, and error recovery.

TCA is used for the AMBLER planetary rover for space exploration and for a mobile manipulator robot.

2.4.3 THEO

The THEO architecture [Mitchell *et al.*, 1990] is intended to provide the underlying inference, representation, and learning mechanisms necessary for increasingly intelligent behaviour. A domain in THEO is expressed as a set of beliefs organized in a knowledge base that has a frame-based representation. Any goal is translated into a belief slot in a frame that has no value, which triggers the inference. The value is obtained using one of the methods associated with the slot. These problem solving methods include inheritance, retrieving default values, and defining a slot's value in terms of other slots. If many methods are available, the system begins searching through them until one is successful. After a problem is solved, THEO stores the solution along with an explanation that justifies it in terms of the beliefs on which it depends. Caching the value of a slot when it is computed avoids recomputing it in the future. Explanations are used for truth maintenance and also as a macro-method for future computation of the slot. Control knowledge is acquired through an inductive learning mechanism that orders the methods available for a slot.

The THEO-Agent [Mitchell, 1990], the main application of THEO, is a robot control architecture for mobile manipulation.

2.4.4 PRODIGY

PRODIGY [Carbonell *et al.*, 1990] is a computational architecture that is aimed at integrating learning and problem solving behaviours. The core of the system is a general-purpose problem solver and planner. A problem is given by an internal state, representing the current state of the world, and a goal state. Domain knowledge is represented in a set of operators and inference rules. The operators are models of the available actions and they specify the effects of the actions under different conditions. Inference rules are used to deduce additional information from the state. PRODIGY searches for a solution using backward chaining means-end analysis by default, but it can be configured to perform other search strategies, including breadth-first search, best-first search, and depth-first iterative deepening. The problem solver is nonlinear and has a very powerful language to express both domain and control knowledge.

PRODIGY uses a least-commitment strategy for every decision in the search process. Control rules are applied at each decision point (to choose goals, operators, etc.) to guide the search. They may express definitive selections or heuristic recommendations. If no control rules are available for a certain decision, the choice is made randomly. Control rules can be hand-coded or automatically acquired. The static module constructs control rules by analyzing the domain description prior to problem solving. Then, an explanation-based learning module examines the problem solving traces and acquires additional rules. All the rules are subject to a dynamic utility analysis that recommends which rules are useful and should be retained. Learning in PRODIGY is combined with problem solving through the automatic acquisition of episodes useful for analogical reasoning, producing abstraction hierarchies, and learning control rules. The acquisition of domain knowledge from the environment is possible both from the user through an apprentice system and from the environment through autonomous learning by experimentation.

PRODIGY has been tested in a variety of domains that include logistics planning, machining and process planning, job-shop schedulling, and computer configuration.

2.5 Multi-Level Architectures

In the research of new methodologies for constructing problem solving systems, an important role has been played by the so-called *multi-level architectures*. In those architectures, the problem solving tasks and the domain models are divided in a set of levels or layers, with different goals. The splitting of the whole intelligent behaviour into several levels, makes multi-level architectures easier to understand, to design and to analyze.

The modelling of different *kinds of knowledge or domain theories* into separate levels provides the architectures with additional *modularity* and *independence* that are always desirable in computational systems for achieving more general architectures [Sierra, 1989]. Both these features have already been studied in data representation tasks through the abstract data types (ADT) within computer science field.

These architectures have been specially developed in the Knowledge-Based paradigm, where another distinction between the domain level (what knowledge is available) and the control level (how to use that knowledge) [Davis and Lenat, 1982; Davis, 1980] is a key feature for a reliable knowledge engineering task. Thus, multi-level architectures can be understood from two dimensions: first, bearing in mind the different *kinds of knowledge or domain theories* (representation), and secondly, from the *domain-control* (syntactic/semantic) insight (see figure 2.9). Most of Knowledge-Based multi-level architectures have been analyzed attending to the second dimension, and they have been called as *meta-level architectures* (as will be explained below).

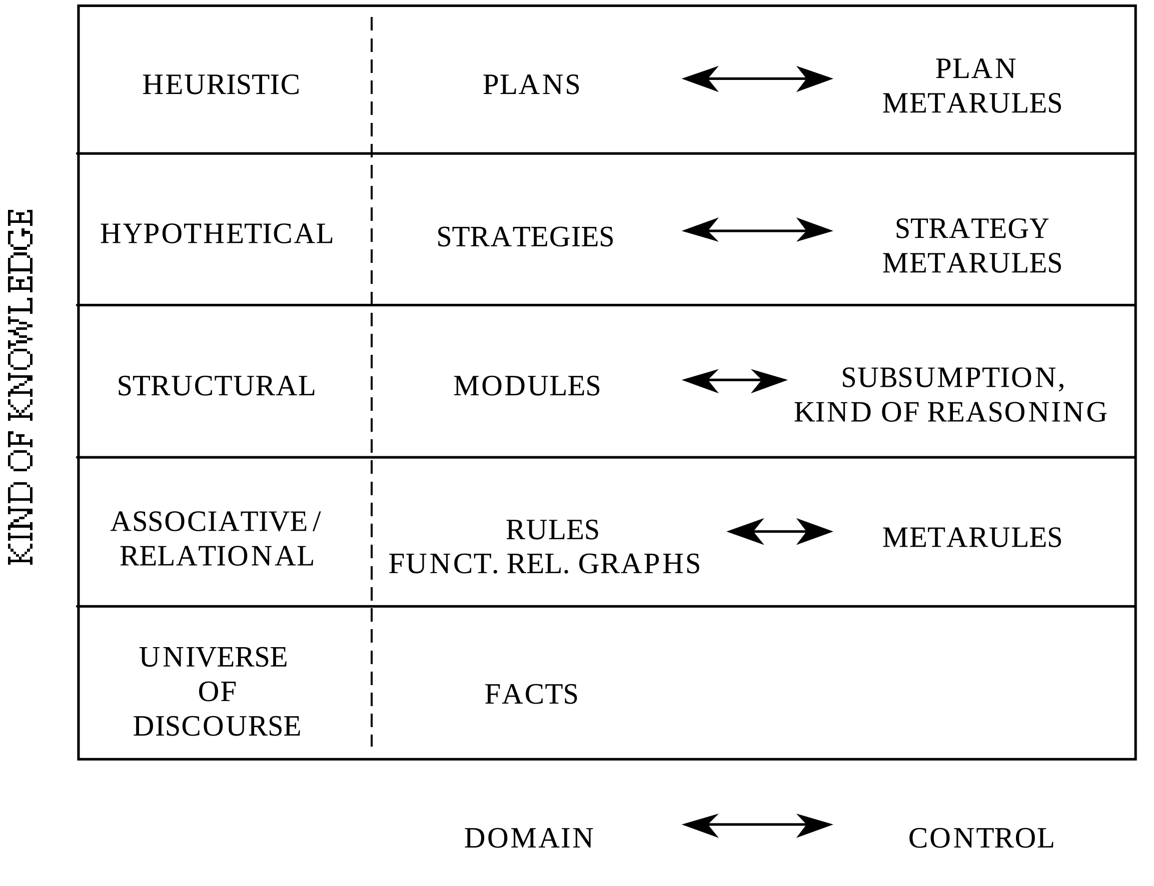


Fig. 2.9. The two dimensions of MILORD [Sierra, 1989]: a KB multi-level architecture

2.5.1 Meta-Level Architectures

TEIRESIAS [Davis and Lenat, 1982; Davis, 1979], that was built-up within the context of MYCIN [Buchanan and Shortlife, 1984], was one of the first systems where two differentiated levels were defined: the *object-level knowledge* (or the domain knowledge) and the *meta-level knowledge* (or the control knowledge).

Meta-level architectures [Van Harmelen, 1991; Russell and Wefald, 1991; Sierra, 1989; Maes and Nardi, 1988; Genesereth, 1983] in general are composed of two levels: *object level* and *meta-level* (see figure 2.10). Each level can be viewed as an independent system with its own representation language, knowledge base and inference mechanism. Knowledge embodied within the object level is domain knowledge. Knowledge embodied within the meta-level is knowledge about the object level knowledge, i.e. meta-knowledge.

Both levels cooperate in problem solving tasks. Inferences that lead the system to problem solving are made at the object level, and the meta-level controls and direct the object level inferences to get an efficient problem solving method. The ability of the meta-level to reason about the problem solving process of the object level has caused that meta-level architectures are also known as *reflective systems* [Maes, 1988], so that they reason about their own internal state. In a meta-level architecture, three features can be distinguished [Van Harmelen, 1991]:

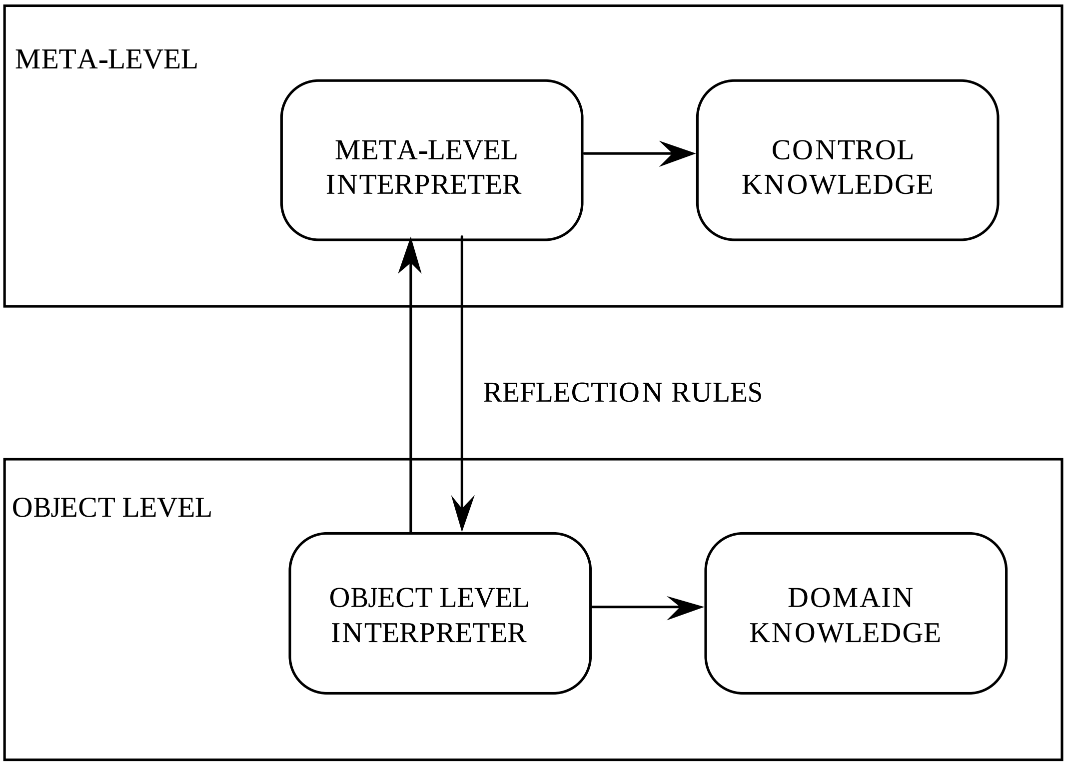


Fig. 2.10. General chart of a meta-level architecture

• The partial view or model (M) that the meta-level has about the object level. For example, the meta-level can only know the set of inputs to object level as a control features, *i.e.*, features that allow the evaluation of the problem solving status at the object level in a certain moment (inference structures, control features, heuristic role annotations, *etc.*).

• A set of *reflection rules* that lead the connection between the two levels[1]. The *upwards reflection rules* referee the object level information that it is needed by the meta-level, according to the partial model M. For example, an upwards reflection rule can be described as a set of information at the object level that becomes a control feature at the meta-level. The *downwards reflection rules* normalize the modifications done by the meta-level over the object level. For example, a downwards reflection rule can explicit a plan built-up at the meta-level that structures the tasks of the object level. The reflection rules ensure a causal connection between the two levels of the architecture.

[1]This concept is related with the *reification* process [Russell and Norvig, 1995], where a predicate or function in first-order logic turns into an object in the language level.

• The mechanism that determines which level is the active level in the reasoning process. Although there are systems that only activate the meta-level in *impasse* situations, others execute major activities at the meta-level, and even can simulate the object level execution. The step in the reasoning process from the object level to the meta-level is known as *meta-level lift up*, and the step from the meta-level to the object level is known as *meta-level reflect down*. The activating mechanism between both levels can be explicit (expressed by means of metarules, for example) or implicit (the inference engine has an execution cycle that alternatively switches from one level to the other).

2.6 Distributed Artificial Intelligence

It is known as *distributed AI* the research, analysis and development of "intelligent communities" that integrate a coordinated set of knowledge-based processes, usually called *agents* (or actors or knowledge sources) that interact either by cooperation, by coexistence or by competition, in order to solve common objectives. Main reasons for distribute an AI system into a multiagent architecture could be enumerated as follows, [Chaib-draa *et al.*, 1992; Castillo and Quintanilla, 1991; Bond and Gasser, 1988; Huhns, 1987]:

• *Geographic distribution* in the domain of application (air traffic control, information systems, robots' system, etc.).
• *Functional decomposition* in a natural way (top down analysis) such in a medical diagnosis, speech recognition system, etc.
• Control distribution in order to get *faster processing speed by means of parallel (concurrent) execution* of agents' work.
• Distribution affords *modularity*, and therefore *reusability* and *extendibility* of the system.
• Processing distribution is a basic strategy to *control the increasing complexity of AI systems*.
• Integration and cooperation of several intelligent agents *(systems) increases the power of the resulting system*.

Also, there are some problems with a distributed architecture [Cammarata *et al.*, 1983]:

• How to maintain the global *coherence* of several agents involved in the architecture ?
• How to *plan* a concrete solution for a given problem ?
• How to *coordinate* agents' communication ?

2.6.1 Models

There are many types of distributed AI architectures which can be grouped in four main classes, [Kirn and Schneider, 1992]: *blackboard systems* (BBS), *supervisory systems* (SVS), *contract nets* (CN), *non explicitly coordinated systems* (NECS):

• *Supervisory Systems* (SVS): its main feature is the existence of a centralized control component. This agent plans all global activities (tasks distribution, cooperative problems resolution, solution's generation, etc.). The supervisory acts like a master (see figure 2.11).

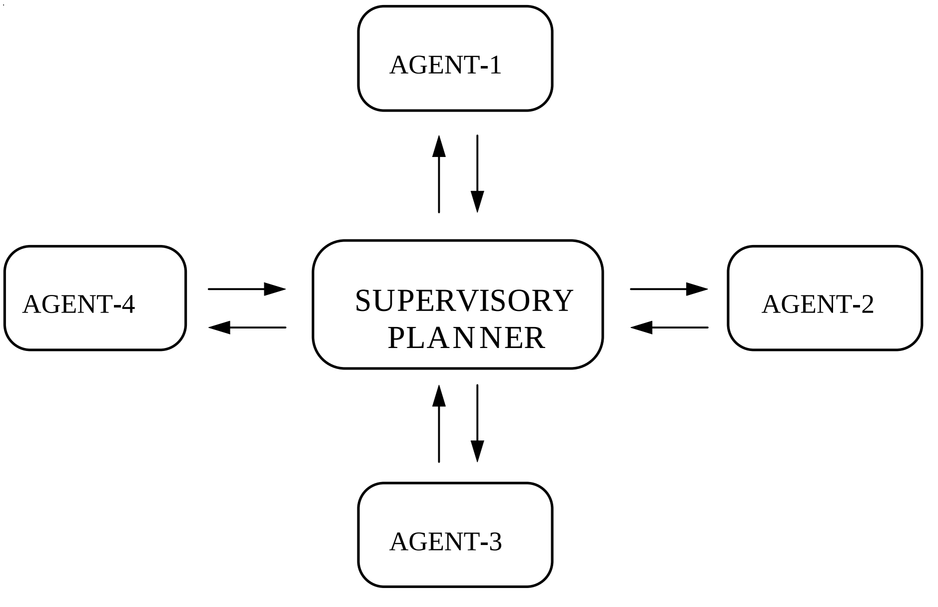


Fig. 2.11. A Supervisory system with 4 agents

• *Blackboard Systems* (BBS): its main characteristic is the blackboard (a centralized data structure that could be accessed by all agents). All steps to the solution have to be documented on the blackboard, which is managed by an intelligent planner (see figure 2.12).

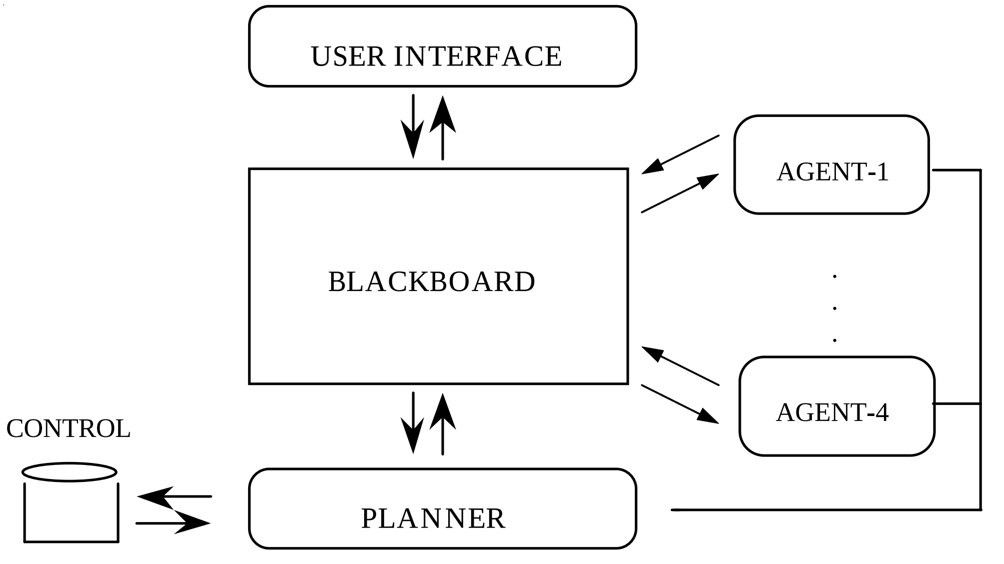


Fig. 2.12. A Blackboard system with 4 agents

• *Contract Nets* (CN): there is not any centralization at all. Agents are completely autonomous. They neither share any resources nor have any global knowledge. They use a contract protocol to cooperate in a flexible way for tasks solving. Usually, the coordination is based in bilateral cooperation. (see figure 2.13).

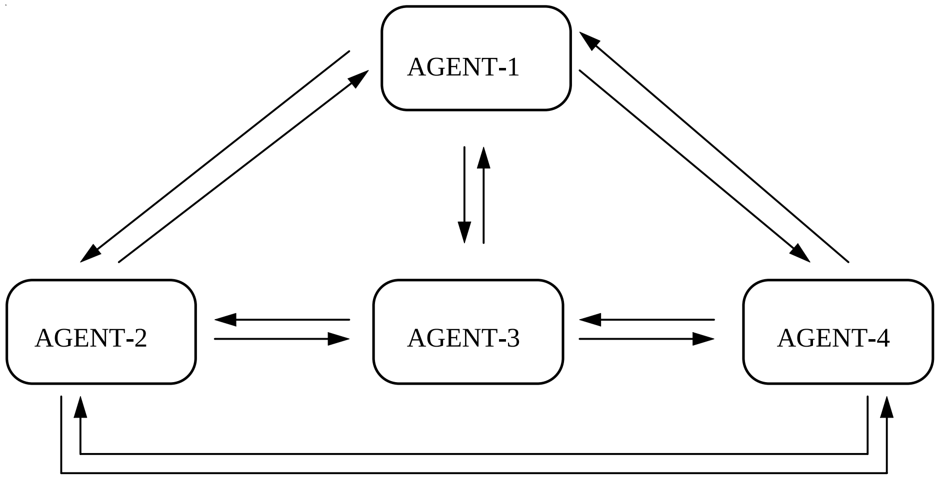


Fig. 2.13. A Contract net with 4 agents

• *Not Explicitly Coordinated Systems* (NECS): They are formed of very complex autonomous agents. If needed, exchange data through a centralized data base. They have any explicit mechanisms for cooperative problem solving such as tasks decomposition, results synthesis, *etc.*. See figure 2.14.

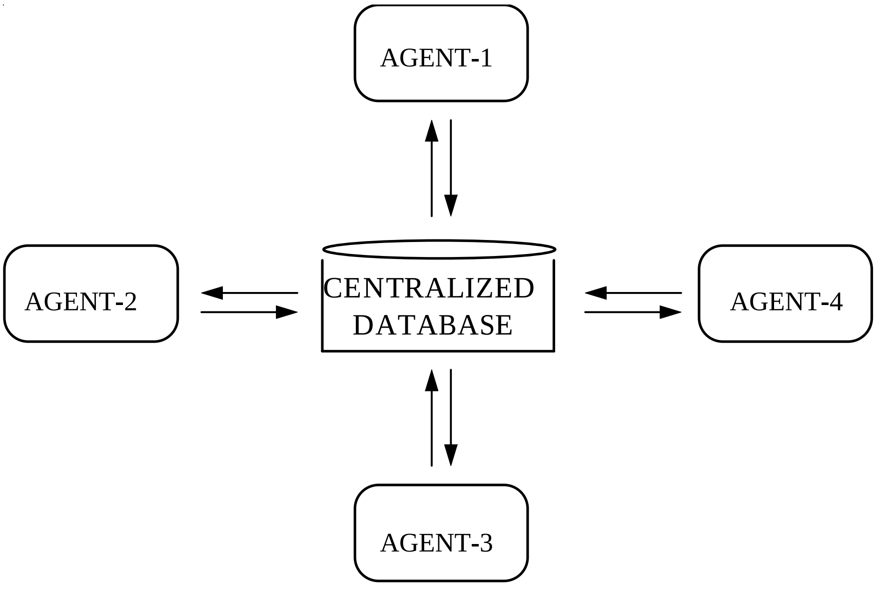


Fig. 2.14. A Not explicitly coordinated system with 4 agents

On the other hand, as suggested in [Bond and Gasser, 1988] Distributed Artificial Intelligence can be divided in two major subfields –disregarding *Parallel AI*, PAI– focusing on historical interests of researchers:

Distributed Problem Solving (DPS) architectures, in which the work of solving a particular problem can be divided among a number of modules, or nodes (agents), that cooperate at the level of dividing and sharing knowledge about the developing solution [Lesser and Corkill, 1987; Smith and Davis, 1981]. Usually, the goals of the different modules (agents) do not interact among them.

Multi-Agent (MA) architectures, where the research is concerned with coordinating intelligent behaviour among a collection of (possible pre-existing) autonomous intelligent "agents" and how they can coordinate their knowledge, goals, skills, and plans jointly to take action or to solve problems. The agents in a multi-agent system may be working toward a single global goal, or toward separate individual goals that interact. The agents, like modules in a DPS system, must share knowledge about problems and solutions., but they must also reason about the processes of coordination among the agents. In MA systems the task of coordination can be quite difficult, for there may be situations (in so-called *open systems*) where there is no possibility for global control, globally consistent knowledge, globally shared goals or global success criteria, or even a global representation of a system [Hewitt, 1986; Hewitt, 1985].

2.6.2 General applications

There are some areas where distributed AI systems have successfully been applied, such as in:

- Air traffic control [Cammarata *et al.*, 1983].
- Robotic systems [Gasser, 1987].
- Man-machine cooperation and office information systems, [Nirenburg and Lesser, 1988].
- Design [Bond, 1989].
- Industrial process control [Roda *et al.*, 1990].
- Medical diagnosis [Gómez and Chandrasekaran, 1981].
- Speech and natural language processing [Fum *et al.*, 1988].

But we have no knowledge about distributed problem solving AI systems applied to wastewater treatment plants.

Chapter 3

DAI-DEPUR: an Integrated Supervisory Multi-level Architecture

3.1 Introduction

The proposed architecture, DAI-DEPUR that stands for *integrated supervisory multi-level distributed system* [Sànchez *et al.*, 1995a; Sànchez *et al.*, 1994b], is the result of previously developed systems, that were oriented to tackle the complexity involved in the control and supervision of WWTPs, during our research work. First, the work was focused on diagnosis or evaluation phase and afterwards on the global supervision and control of the system:

- Knowledge-based diagnostic and management system [Serra *et al.*, 1994; Belanche *et al.*, 1992b; Sànchez, 1991].

- Real-time supervisory system [Serra *et al.*, 1995; Sànchez *et al.*, 1994a; Serra, 1993].

The architecture is designed to overcome some of usual troubles of KBS and those of conventional control systems, as explained in previous chapters. The integrated supervisory multi-level distributed system (as shown in Figure 3.1) is formed of several interacting subsystems or agents that can be executed in parallel processing (as the Supervisory–KBS agent, the Case-Based Reasoning and Learning agent, the Numerical Control Knowledge module, *etc.*). The architecture is also *integrated* and *multi-level* (as explained in section 3.3).

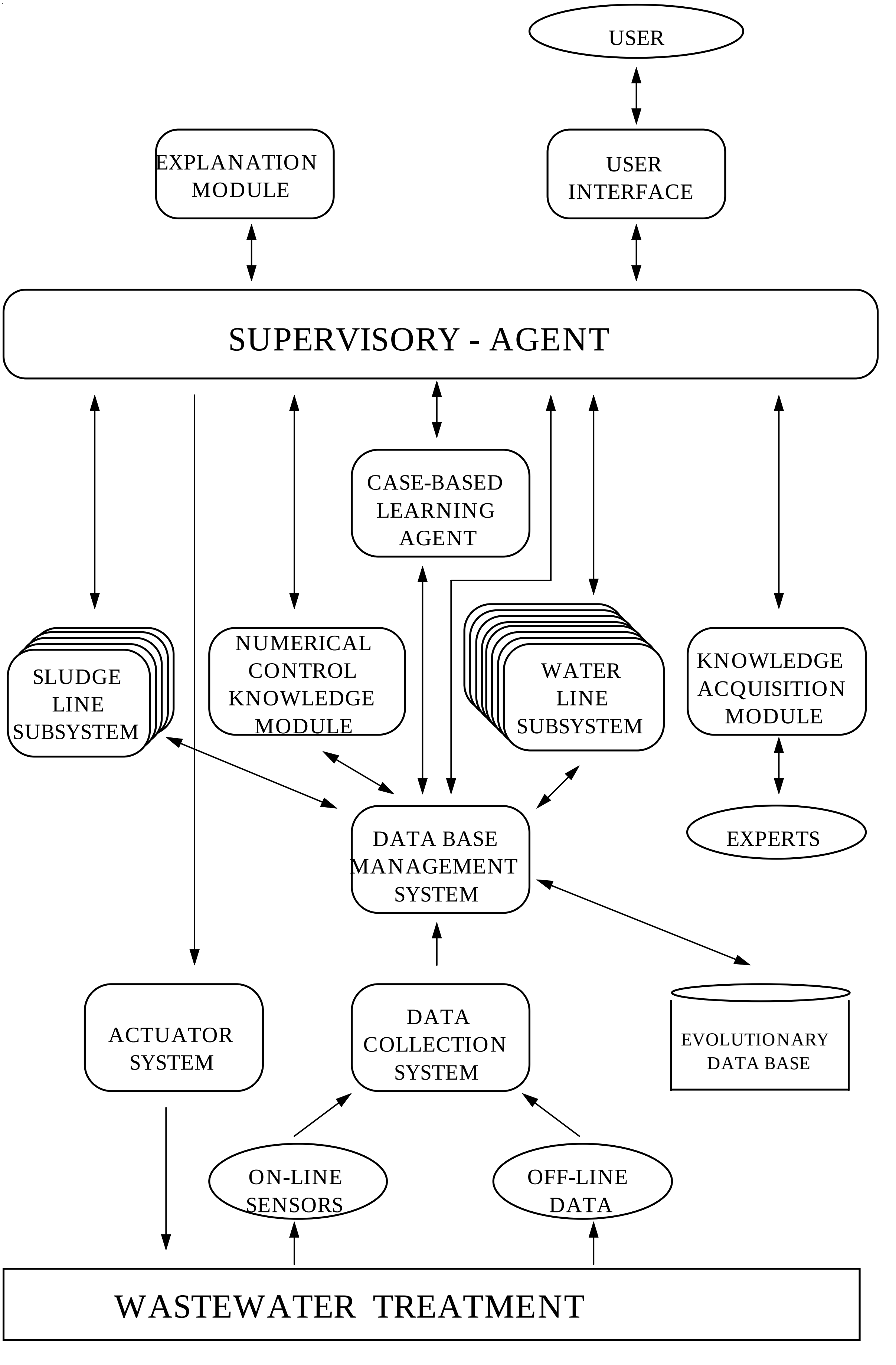

USER
EXPLANATION MODULE
USER INTERFACE
SUPERVISORY - AGENT
CASE-BASED LEARNING AGENT
SLUDGE LINE SUBSYSTEM
NUMERICAL CONTROL KNOWLEDGE MODULE
WATER LINE SUBSYSTEM
KNOWLEDGE ACQUISITION MODULE
DATA BASE MANAGEMENT SYSTEM
EXPERTS
ACTUATOR SYSTEM
DATA COLLECTION SYSTEM
EVOLUTIONARY DATA BASE
ON-LINE SENSORS
OFF-LINE DATA
WASTEWATER TREATMENT

Fig. 3.1. DAI-DEPUR architecture

The main reason to choose a Supervisory Distributed Problem Solving system among the different Distributed Architecture models (see 2.6.1) is mainly because in the domain of WWTP the outstanding task is to identify *situations*, that is operational working states of the plant described by means of the relevant attributes of the system, and among those, there is a set of *abnormal situations* that could be easily catalogued as *Storm*, *Bulking*, *Toxic load*, *etc*. We call them *usual abnormal situations*, and they may be treated with a predetermined plan or strategy, in a more efficiently way than with other types of DAI architectures as Blackboard Systems or Contract Nets. Another reason is that the Supervisory Systems ensure, by means of the supervisory-planner agent, an easy and reliable planning, coordination and coherence among all agents, although this centralization diminish the flexibility of the architecture, the agents autonomy and agents global knowledge.

The Supervisory System recognizes predefined *usual abnormal situations* and chooses the right strategy, in addition to specific experiential situations occurred in the concrete WWTP under control (*unusual abnormal situations*), in order to keep the process controlled, or if *normal situation* has been detected then the automatic numerical control is maintained or activated. Normal means that the WWTP is correctly operating, and so, the contaminant's levels of the effluent water are under the limits of environmental laws. Thus, it is clear that there is room for a good cooperation among *classical control methods* (based on numerical algorithms), *expert control* (based in predefined plans or strategies) and *experiential control* (based in previous solved cases).

In the next sections of this chapter, an insight of the main features of the architecture will be given. In next chapters, each DAI-DEPUR level will be described in detail.

3.2 Knowledge-Level Analysis of WWTP

Attending to the components of expertise approach to analyse the WWTP domain, we will describe the three perspectives: domain models or theories, tasks, and methods.

We briefly describe the *models* used in the architecture at each level. The data level is modelled with the on-line data coming from WWTP sensors, the calculated and inferred data, and off-line data required from WWTP operators. At the knowledge/expertise level there are three domain models: a *functional expert (symptom-cause)* model, an *associative experiential (cases)* model, and a *numerical (control algorithm)* model. At the next level, there are the set of expert and experienced situations. Finally, at the plan level, there are the strategies or actuations: experiential/expert actuation and numerical control actuation.

The main *tasks* involved in the overall management process in a WWTP are: system evaluation to get the main values and parameters of the WWTP operation; diagnosis, to determine which is the current working situation of the WWTP; adaptation, which is an optional task sometimes performed to update the knowledge/expertise level; supervision, to identify the current situation between expert general or experiential specific situations provided by the diagnosis task; the validation of the supervision task (identified situation, proposed solution); and finally the actuation, based on the proposed solution to restore the WWTP operation to the normal situation.

The *problem solving methods* used to perform the tasks are: the evaluation task is performed by means of the on-line and off-line data gathering, the diagnosis task is implemented both with rule-based reasoning and case-based reasoning. Case-based learning and knowledge acquisition perform the adaptation task. The supervision is implemented with a combination method. The validation of the proposed solution is achieved by means of the operator's validation. The actuation over the WWTP is performed by a numerical control actuation or by an experiential/expert actuation.

3.3 Integrated Multi-level Architecture

DAI-DEPUR is an *integrated* architecture because it joins in a single system several cognitive tasks and techniques such as learning, reasoning, knowledge acquisition, distributed problem solving, *etc*. Furthermore, focusing on the *knowledge/expertise level*, [Steels, 1990], there is the integration of three kinds of knowledge: numerical control knowledge, expert general knowledge and experiential specific knowledge [Sànchez *et al.*, 1995b].

The need of different reasoning techniques integrated within a single system for solving complex real-world problems has been recently recognized by the AI community [Dutta and Bonissone, 1993]. Only there was an early precedent that has not much been taken into account by the AI researchers [Aikins, 1983], where it was proposed to apply prototypical knowledge (cases or experiences) to control (meta-reasoning) the rule-based reasoning of expert systems.

In the AI literature there have been some approaches to combine the case-based reasoning and the rule-based reasoning [Cuda *et al.*, 1994; Golding and Rosenbloom, 1991; Branting, 1991; Dutta and Bonissone, 1991; Rissland and Skalak, 1989] from a complementary point of view between both kinds of reasoning.

Also, DAI-DEPUR architecture is *multi-level*, providing independence to all the levels. Taking into account the domain theory (models), it can be structured as a four-level architecture: data level, expertise level, situations level and plans level.

Data level: On-line data gathered from sensors, calculated and inferred data, and off-line values provided by the operator as laboratory analysis, subjective information, *etc.*

Knowledge/Expertise level: Modelled by three paradigms or approaches: numerical control knowledge, expert general knowledge and experiential specific knowledge.

Situations level: The generic global operating situation of a plant is obtained by combination of its several local subsystem situations. Also, this generic global situation and the specific situation provided by the experiential knowledge are combined.

Plans level: At this level, the identified whole situation, the previous similar solutions as well as the predefined (canned) plans are taken into account to propose a first solution, that has to be validated against the operator, who can modify the proposed plan. Then, an arranged plan can be executed to cope with the current operating situation of the plant. Plans are a sequence of actions to be taken in order to restore the good WWTP operation and performance.

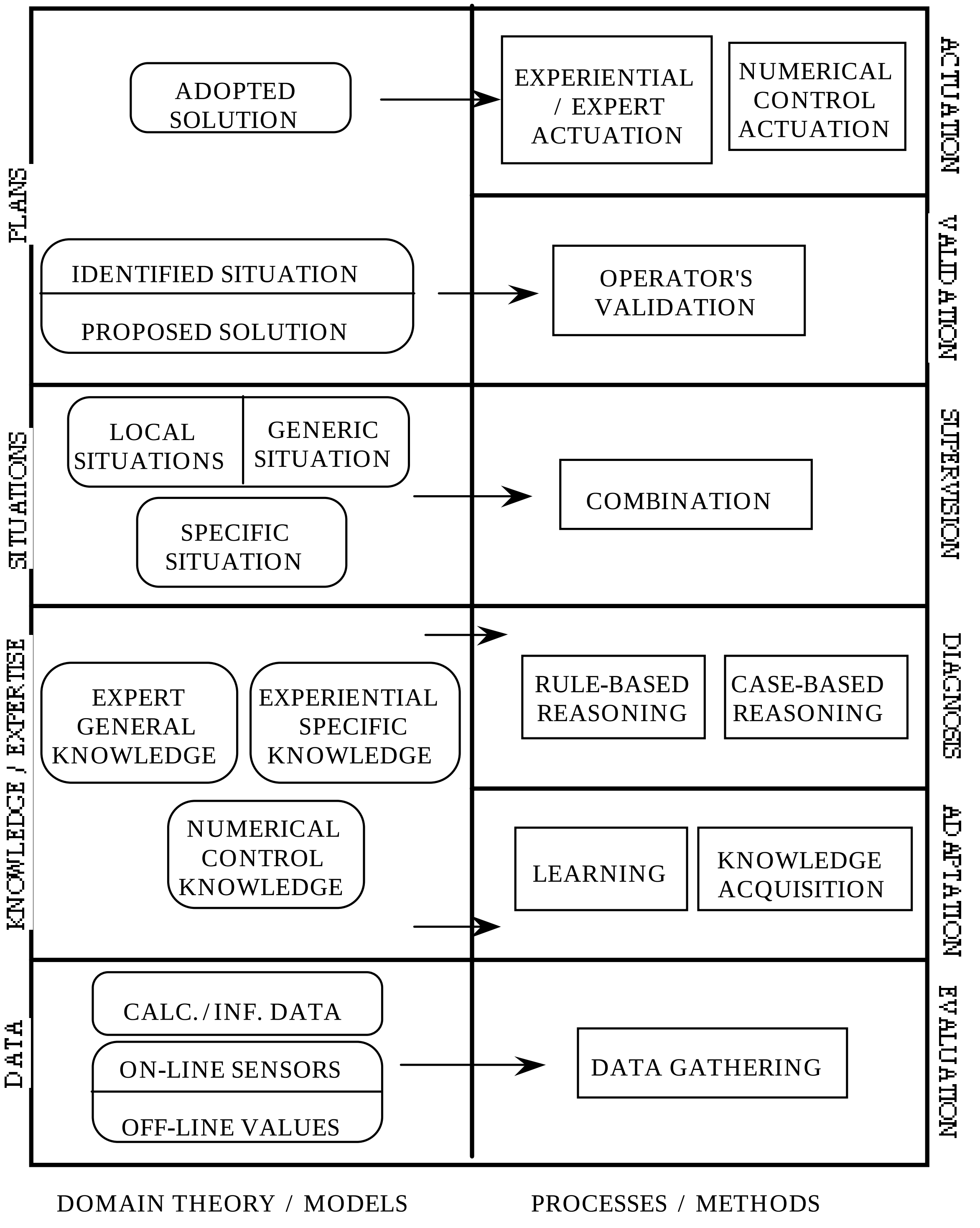


Fig. 3.2. Integrated supervisory multi-level architecture

On the other hand, considering the processes acting over the models (methods) the architecture can be decomposed in a six-phase processes: evaluation process, adaptation process, diagnosis process, supervision process, validation process and actuation process. The system activates a new supervisory cycle at fixed intervals of time.

Evaluation process: For this purpose it is necessary to know some values for certain variables of the process. All this data can be extracted from the evolutionary Data Base, fitted either with the on-line sensors values coming from the data collecting systems or with some other features provided by the operator like a laboratory analysis, qualitative observation, *etc.*, as well as the inferred and computed data.

Adaptation process: This is a process that is sometimes performed either by dynamic learning from past proposed solutions and its efficiency –that can update the Case library– or by acquiring some new knowledge from (new) experts or (new) sources through classification techniques.

Diagnosis process: In a new cycle the Supervisory agent activates the Knowledge-based agents to diagnose the state of the different subsystems of the plant by means of rule-based reasoning. At the same time in the diagnosis phase the Case-Based Reasoning and Learning agent (CBRL) is activated to retrieve similar cases recorded in the Case library. This means that can be implemented the concurrent execution of all agents involved. In the next step, is updated the most similar case in order to adapt it to the current situation of the plant. For this task, the system needs to access to the Data Base. The results are communicated to the Supervisory agent.

Supervision process: The Supervisory agent combines all information coming from the several KBS agents (*general knowledge*) and from the CBRL agent (*specific knowledge*) to infer the current global situation of the plant and the suggested actions to be taken. It sends this information to the operator through the User Interface module.

Validation process: The system can be inquired by the operator in some ways as asking for explanations, retrieving certain values, *etc.* The Supervisory agent waits for the operator's validation of actions to be taken in order to update the current working state of the plant.

Actuation process: The Supervisory agent recognizes *situations* and uses the right strategy or plan, in order to keep the process controlled or if *normal situation* has been detected, then, the automatic numerical control is maintained or activated.

If there are on-line actuators, the plant can be automatically updated through the Actuator system. If not, manual operation is required.

3.4 Distributed Problem Solving

DAI-DEPUR is a *distributed problem solving* architecture (DPS) so that is formed of several cooperation subsystems where the work of solving a particular problem is divided among a number of modules, or nodes (agents) that can be executed in parallel. For instance, the supervisory agent, the case-based reasoning agent, primary settler-KBS agent, biological reactor-KBS agent, *etc.* Distribution criteria are based on spatial and semantic distances, [Bond and Gasser, 1988]. In a WWTP there are some subsystems that are in spatially distributed locations (primary settlers, secondary-settlers, biological reactor, *etc.*). Thus, all of these knowledge-based agents are specialized ones, that focus on different aspects of the system. Each one has its own knowledge base, that has been obtained by means of the knowledge acquisition module:

• The Water line subsystem formed of: Screen–KBS, Grit removal–KBS, Primary settler–KBS, Biological reactors–KBS, Secondary settler–KBS, Chlorination–KBS, Recirculation–KBS, which will be described in 5.2.2.2.

• The Sludge line subsystem composed by: Waste–KBS, Thickening–KBS, Anaerobic treatment–KBS, Drying–KBS, which will be described in 5.2.2.2.

• The Supervisory–KBS agent, which will be described in 6.3.

The other nodes of the architecture are:

• The Case-Based Learning and Reasoning agent, which will be detailed in 5.2.3 and 5.4.2.
• The Numerical Control Knowledge module, that will be described in 5.2.1.
• The Knowledge Acquisition module, which will be detailed in 5.4.1.1.
• The Actuator system, which will be described in 7.4.
• The Data Base Management system, which will be explained in chapters 4 and 9.
• The Data Collecting system, which will be explained in chapters 4 and 9.
• The User Interface module, which will be explained in chapters 4 and 9.

- The Explanation module, which will be explained in chapter 7 and 9.

The proposed architecture is implemented using some existing tools as G2[1] [Gensym, 1992; Gensym, 1990], LINNEO^{+}[2] [Béjar, 1995; Béjar *et al.*, 1994], GAR[3] [Riaño, 1994], and Sun Common Lisp[4] [Sun Microsystems, 1990]. See appendix G for a brief description of these tools.

The architecture implementation has been specially focused on Manresa's WWTP, but also the experiences both from a slightly different WWTP studied in Cassà de la Selva-Llagostera, and the pilot scale WWTP constructed have been taken into account in DAI-DEPUR development.

---

[1] G2 is a real time expert systems shell, from Gensym Corporation
[2] LINNEO^{+} is a semi automatic knowledge acquisition tool developed at LSI dept.
[3] GAR is an automatic rule generation tool developed at LSI dept.
[4] Sun Common Lisp is the Common Lisp dialect from Sun Microsystems

Chapter 4

The Data Level

The sight always must learn from the mind.
Johannes Keppler

4.1 Domain Models

At the data level, the domain theory is represented by means of three information models: the *on-line data* coming from WWTP sensors, the *calculated and inferred data* by the system, and the *off-line values* provided by the operator. In next subsections, all these data are detailed. In the appendix A, more accurate definitions can be found.

All the ranges of these data are true for the application developed for the Manresa's WWTP. Other range values are available for the other applications (Cassà de la Selva-LLagostera's WWTP, pilot scale WWTP).

4.1.1 On-line Data

There are many signals captured by the data acquisition interface as explained in 4.1.1.1. Many of them are information about the status of pumps engine (on / off), or the status of aeration-turbines engine (on / off), or the status of automatic grids engine (on / off), or about the status of electrovalves engine (on / off), *etc*. Excluding these mechanical-electrical signals, the main outstanding on-line data gathered by DAI-DEPUR are:

- Inflow: numerical measure of the water flowing into the WWTP. Possible values are:
  - low (< 28000 m^3/ day)
  - slightly low (between 28000 and 35000 m^3/ day)
  - normal (between 35000 and 45000 m^3/ day)
  - high (> 45000 m^3/ day)

- Input-pH: numerical measure of the pH of the input water to the WWTP. Possible values:
  - low (between 0 and 6)
  - normal (between 6 and 8.5)
  - high (between 8.5 and 14)

- Dissolved-Oxygen (DO): numerical measure of the concentration of the dissolved oxygen in the biological reactor (basins). Possible values are:
  - low (between 0 and 0.5 ppm)
  - normal (between 0.5 and 2.5 ppm)
  - slightly high (between 2.5 and 4 ppm)
  - high (> 4 ppm)

- Recirculation-flow: numerical measure of the recirculated flow of sludge. Possible values are:
  - low (< 30000 m^3/ day)
  - normal (between 30000 and 50000 m^3/ day)
  - high (> 50000 m^3/ day)

- Wasting-flow: numerical measure of the purged flow of sludge. Possible values are:
  - low (< 450 m^3/ day)
  - normal (between 450 and 600 m^3/ day)
  - high (> 600 m^3/ day)

In other WWTPs it can be found on-line sensors to obtain some of the following data: Oxidation-Reduction Potential (ORP), Turbidity, Water temperature, Ammonia, Nitrate, Nitrite, *etc*.

4.1.1.1 External interface

The *Data collection module* developed for Manresa's WWTP application gathers the values of the status of turbines, pumps, automatic grids, *etc*., from the control panel of the plant under supervision, with 6 PLC's (SISTEL 8512). These 96 digital signals are transmitted through RS-422 to the monitoring computer (PC) for evaluation process. It also receives 9 analogical signals (inflow, wasting flow, recirculation flow, biogas produced, DO-line-1, DO-line-2, pH, temperature-digester-1 and temperature-digester-2) converted by an AD/DA card. The data acquisition interface is connected to the main computer (SUN Sparc station) where all the other agents and processes are running (see figure 4.1).

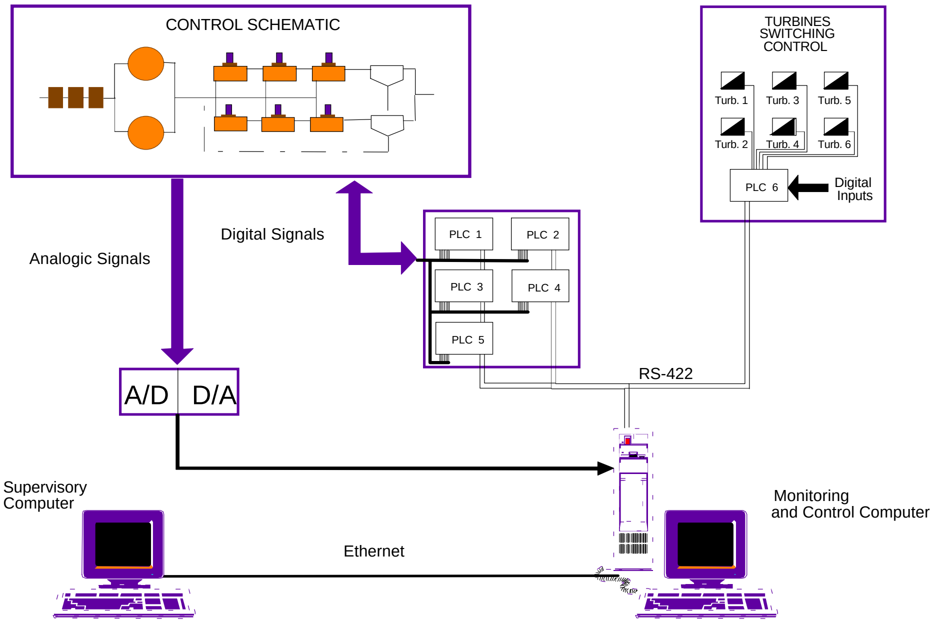


Fig. 4.1. On-line external interface

In other WWTPs, such as Cassà de la Selva-Llagostera, the data acquisition interface module has to be updated to include different sensor information from the WWTP to DAI-DEPUR.

Currently, there are some experiments using a pilot scale plant, that is already connected to a monitoring computer through a data acquisition interface. See chapter 8.

4.1.2 Calculated and Inferred Data

The main outstanding inferred and calculated data by DAI-DEPUR are:

• Bridge-rotation-band: qualitative information about the status of the rotation band of the bridge. There are two possible values:
normal
broken

• Primary-settler-sludge-exit: qualitative information about the status of the primary settler sludge exit. Possible values are:
normal
insufficient

• Turbines-schedulling: qualitative information about the status of the turbines' schedulling. Possible values are:
following-schedulling
not-following-schedulling
broken
overloading

• Solids-Retention-Time (SRT): numerical measure of the mean residence time of the Suspended Solids in the biological process. Possible values are:
young (< 5 days)
normal (between 5 and 8 days)
old (> 8 days)

• Sludge-Volume-Index (SVI): numerical measure of the sludge sedimentation ability. Possible values are:
low (< 80 g / ml)
normal (between 80 and 150 g / ml)
high (> 150 g / ml)

- Removal-efficiencies (%): percentage of removal efficiency of the main water quality variables such as COD, BOD, Suspended Solids (SS), *etc*.

4.1.3 Off-line Data

The data that cannot be either obtained on-line, or calculated or inferred by DAI-DEPUR, must be provided off-line, by the WWTP operators, through the user interface such as observational information, laboratory analysis, microscopical information, *etc*. These data are:

- Bubbles-in-primary-settlers: qualitative information about the absence or the presence of bubbles in the primary settlers, thus indicating a septic water. Possible values are:
  - true
  - false

- Bubbles-in-clarifiers: qualitative information about the absence or the presence of bubbles in the clarifiers (secondary settlers), thus indicating denitrification. Possible values are:
  - true
  - false

- Bioreactor-foam: qualitative information about the presence / absence of colour in the bioreactor foams. Possible values are:
  - brown (indicating abundance filamentous bacteria presence)
  - white (indicating tensoactive presence or a high F / M ratio)
  - no-foam (normal water conditions)

- Water-odour: qualitative information about the presence or the absence of water odour. Possible values are:
  - true (possible industrial waste)
  - false

• Water-colour: qualitative information about the water colour. Possible values are:

red (indicating some blood concentration in the water))
dark (reflecting industrial water)
normal

• Environmental-temperature: numerical measure of the environmental temperature. Possible values are:

low (< 5º C)
normal (between 5º C and 27º C)
high (> 27º C)

• Input-sulphurs: numerical measure about the presence or the absence of sulphurs in the inflow. Possible values are:

normal (between 0 and 0.5 ppm)
high (> 0.5 ppm)

• Input-heavy-metal-concentration (based on Zn): numerical measure of the concentration of metals (Zn) in the inflow. Possible values are:

normal (between 0 and 4 ppm)
high (> 4 ppm)

• Input-Suspended-Solids (I-SS): numerical measure of the Suspended Solids presents at the input of the WWTP. Possible values are:

low (< 150 mg/L)
normal (between 150 and 300 mg/L)
high (> 300 mg/L)

• Input-Chemical-Oxygen-Demand (I-COD): numerical measure that reflects the quantity of substrate in the inflow water at the input of the WWTP. Possible values are:

low (< 150 mg/L)
normal (between 150 and 400 mg/L)
high (> 400 mg/L)

• Input-Biological-Oxygen-Demand (I-BOD_5): numerical measure that reflects the quantity of substrate in the inflow water at the input of the WWTP. Possible values are:

low (< 100 mg/L)
normal (between 100 and 350 mg/L)
high (> 350 mg/L)

• Bioreactor-heavy-metal-concentration (based on Zn): numerical measure of the concentration of metals (Zn) at the basins. Possible values are:

normal (between 0 and 30 ppm)
high (> 30 ppm)

• Bioreactor-Suspended-Solids (B-SS): numerical measure of the Suspended Solids presents at the basins. Possible values are:

low (< 1750 mg/L)
normal (between 1750 and 3000 mg/L)
high (> 3000 mg/L)

• Bioreactor-Volatile-Suspended-Solids (B-VSS): numerical measure of the Volatile Suspended Solids presents in the bioreactor. Possible values are:

low (< 1500 mg/L)
normal (between 1500 and 2500 mg/L)
high (> 2500 mg/L)

• Bioreactor-pH: numerical measure of the pH in the bioreactor mixed liquor. Possible values:

low (between 0 and 5.0)
normal (between 5.0 and 8.5)
high (between 8.5 and 14)

• Recirculation-Suspended-Solids (R-SS): numerical measure of the Suspended Solids presents in the recirculation flow. Possible values are:

low (< 3750 mg/L)
normal (between 3750 and 6500 mg/L)
high (> 6500 mg/L)

• Recirculation-Volatile-Suspended-Solids (R-VSS): numerical measure of the Volatile Suspended Solids presents in the recirculation flow. Possible values are:

low (< 3500 mg/L)
normal (between 3500 and 6000 mg/L)
high (> 6000 mg/L)

• Output-Suspended-Solids (O-SS): numerical measure of the Suspended Solids presents at the output of the WWTP. Possible values are:

normal (< 30 mg/L)
high (_ 30 mg/L)

• Output-Volatile-Suspended-Solids (O-VSS): numerical measure of the Volatile Suspended Solids presents at the output of the WWTP. Possible values are:

normal (< 25 mg/L)
high (_ 25 mg/L)

• Output-Chemical-Oxygen-Demand (O-COD): numerical measure that reflects the quantity of substrate in the water at the output of the WWTP. Possible values are:

normal (< 75 mg/L)
high (_ 75 mg/L)

• Output-Biological-Oxygen-Demand (O-BOD$_5$): numerical measure that reflects the quantity of substrate in the water at the output of the WWTP. Possible values are:

normal (< 35 mg/L)
high (_ 35 mg/L)

Other off-line signals taken into account in some WWTPs can be: Total Organic Carbon (TOC), Alkalinity, Grease, Chloride, *etc.*

4.1.3.1 Microbiological information

Microbiological information is a crucial off-line qualitative information. In the secondary treatment there are two main kinds of microorganisms attending to their importance in the biological process: *protozoa* and *filamentous bacteria*.

Protozoa

Protozoa use the bacteria community as a nutrient, and it is widely demonstrated [Pujol *et al.*, 1990; Moro, 1984] that they play an important role both in the elimination of the Biological Oxygen Demand (BOD) and in the removal of the Suspended Solids (SS). Main contributions of protozoa to the cleansing process are:

- They separate bacteria from the effluent during the sedimentation process
- They capture disperse bacteria in the absorption process and sediment them
- They diminish the quantity of disperse bacteria, due to their predatory role
- They increase the substrate / biomass ratio (F / M), so they diminish the quantity of bacteria
- They can directly eliminate substrate
- They eliminate pathogen microorganisms

A compilation from different observations [Vedry, 1988] allows us to state the following relationships between the state of the sludge and the presence of protozoa:

Normal sludge: *Vorticella, Epistylis, Aspidisca, Opercularia, Zoothamnium, Carchesium, Euplates, Tokophrya, Podophrya, Acineta.*

Transition sludge: *Litonotous, Loxophilum, Chilodonella, Oxytricha, Amœba.*

Abnormal sludge: *Bodo, Cercobodo, Oikamonas, Paramecium, Vahlkampfia, Metopus, Cænomorpha.*

In general, it can be stated that as much larger is the quantity of protozoa, the quality of effluent water better is.

Filamentous Microorganisms

On the other hand, the presence of *filamentous microorganisms* in the Mixed Liquor Volatile Suspended Solids (MLVSS) is commonly related in the literature and experimentally verified, to some *usual abnormal situations* (*bulking*) [Jenkins *et al.*, 1993; Wanner *et al.*, 1987; Chudoba, 1985; Strom and Jenkins, 1984] as it is described in table 4.1.

Unfortunately, within the periods (90/91, 94) mainly studied at the Manresa's WWTP, there was not many bulking situations, that could have been of great interest to identify the local and particular microorganisms that grow in the concrete plant, because they are very sensitive to the water and environmental conditions. In fact, all this microbiological information becomes a *functional (symptom-cause) model* at this level.

POSSIBLE SITUATION	KIND OF MICROORGANISM
Low dissolved oxygen	1701, S. Natans, H, Hydrosis
Low F/M (low organic load)	M. Parvicella, Nocardia, H. Hydrossis, 021N, 0041, 0092, 0581, 0961, 0803
Septic residuals, Sulphurs	Thiotrix, Beggiatoa, 021N
Nutrient deficiency	Thiotrix, 021N, 0041

Table 4.1. Relationship between filamentous microorganisms and working situations of WWTP

In order to identify what kind of filamentous microorganisms are present in the Mixed Liquor Volatile Suspended Solids (MLVSS), it is required qualitative information about some microorganisms features observed with a microscopy:

- Filaments-branching: qualitative information about the ramifications of the filaments. Possible values are:
  - absence
  - false branching
  - true short branching
  - true long branching

- Filaments-motility: qualitative information about the presence/absence of motility in the filaments. Possible values are:

true
false

• Sulphur-granules: qualitative information about the presence/absence of sulphur granules in the microorganisms. Possible values are:
true
false

• Gram-staining: qualitative information about the result of the Gram-staining of the microorganisms. Possible values are:
positive
negative

• Neisser-staining: qualitative information about the result of the Neisser-staining of the microorganisms. Possible values are:
positive
negative

• Filaments-shape: qualitative information about the shape of the filaments. Possible values are:
straight filaments
coiled filaments
smoothly-curved filaments
chain of cells

• Cell-septa (cross-walls): qualitative information about the visibility of cell septa of the microorganisms. Possible values are:
good visibility
bad visibility

• Sheath-presence: qualitative information about the presence/absence of a sheath in the filaments. Possible values are:
true
false

• Attached-growth: qualitative information about the presence/absence of attached epiphytic bacteria around the filaments. Possible values are:

true
false

• Filaments-transparency: qualitative information about the transparency of the filaments. Possible values are:
true
false

• Cell-shape: qualitative information about the shape of the microorganisms. Possible values are:
round-ended rauds shape
square shape
oval shape
barrel shape

• Cell-diameter: numeric value of the diameter of the cells. Values are expressed in µm.

4.1.3.2 Off-line data interface

All previous off-line information is gathered into DAI-DEPUR through the user interface, by means of some type-in boxes, buttons and menus of the user interface (see figure 4.2).

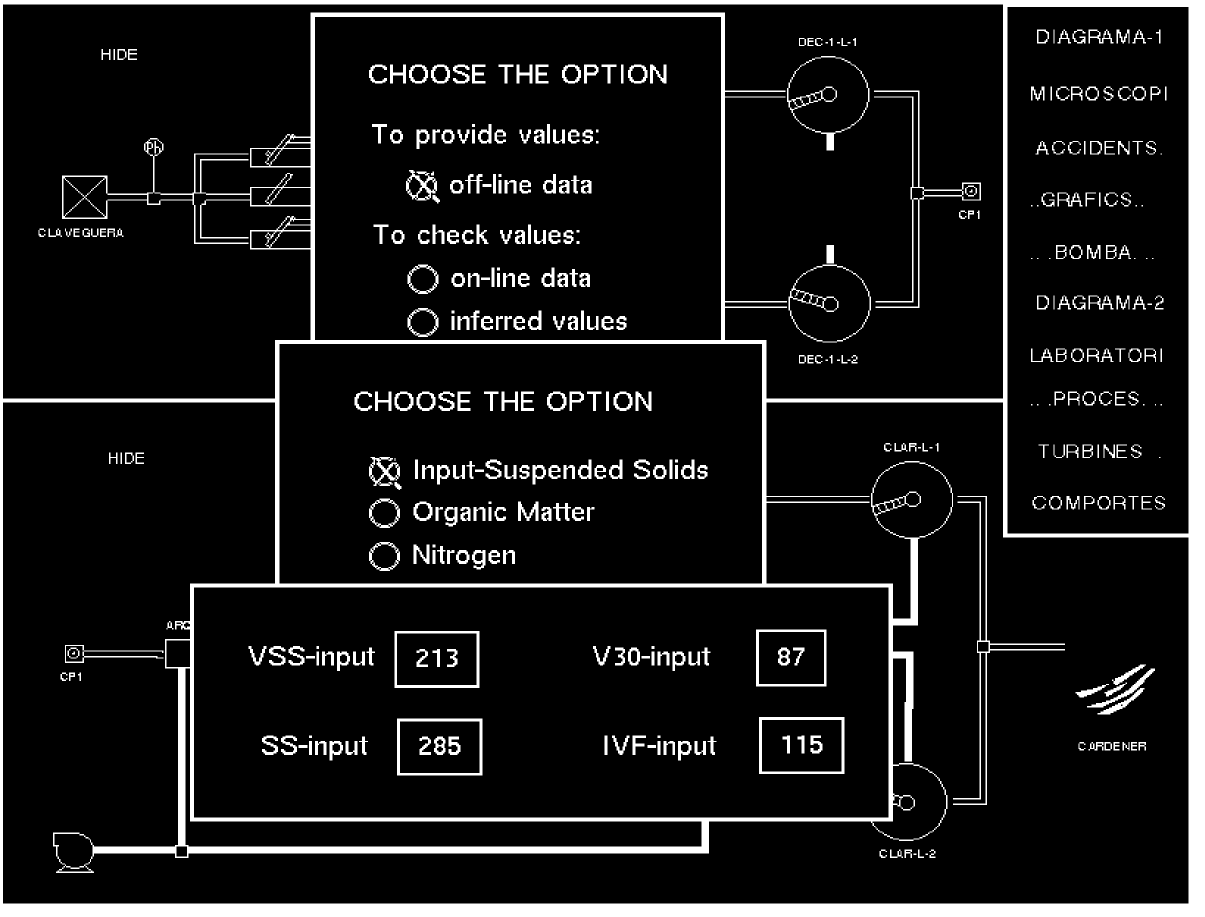


Fig. 4.2. Off-line data interface

4.2 System Evaluation Task

The main task at this level is the evaluation of the WWTP under supervision to get the basic information that will allow DAI-DEPUR to diagnosis -in an upper level– the working situation of the WWTP, to supervise, and to actuate in order to restore the good operational state of the WWTP.

As it has been explained in previous section 4.1, there is much off-line information to be gathered by the WWTP supervisory systems. This is due to the fact that most WWTP are not very automated. First, there are much measures for which technology have not still developed on-line sensors. On the other hand, some on-line sensors developed are not very reliable. This last feature introduces the problem of the *uncertainty* or *approximate information* [López de Mántaras, 1990; Smets *et al.*, 1988] into the supervision process. Moreover, the uncertainty can be originated by subjective information provided by the operators and / or experts, and sometimes, the available information about some parameters can be incomplete [Bonissone and Tong, 1985]. To solve that problem, DAI-DEPUR architecture gives the possibility that each attribute or parameter or fact, may have associated a fuzzy label value through a linguistic approximation [Bonissone *et al.*, 1990; Bonissone and Decker, 1985; Bonissone, 1979] to the fuzzy sets theory [Zadeh, 1983; Zadeh, 1979].

WWTPs are *dynamic* systems that change their behaviour over the time. A good operational state of WWTPs is mainly achieved when the balance between substrate and biomass (F/M) is guaranteed. This balance is very sensitive to the changes in the process operating conditions. Luckily, the dynamics of the WWTPs is enough slow to deaden these perturbations, and not to cause great damage to the environment. This delay in the WWTP information implies than it is not only important the information about the *current state* of the WWTP, but the *previous states* of WWTP working are also crucial. A good domain modelling has to take into account both kinds of information: *current-input* data and *previous-historical* data.

This feature suggests a role for temporal reasoning approach. DAI-DEPUR uses the temporal information management provided by the G2 shell, that allows it to reason taking into account the temporal values recorded in the Data Base of the same variables. Also, it makes use of the two kinds of information above mentioned, to capture these temporal features:

- *Current-input data*, such as the environmental-temperature, wastewater inflow, Input-Chemical-Oxygen-Demand (I-COD), Input-Suspended-Solids (I-SS), Input-metal-concentration, Sludge-Volumetric-Index (SVI), Bubbles-in-primary-settlers, Aeration-turbines-state, *etc.*

- *Previous-historical data*, such as the Outflow-Chemical-Oxygen-Demand (O-COD), Output-Suspended-Solids (O-SS), Dissolved-Oxygen (DO), Water-odour, Bioreactor-foam, Bioreactor-Suspended-Solids (B-SS), Bioreactor-Volatile-Suspended-Solids (B-VSS), Recirculation-Suspended-Solids (R-SS), Recirculation-Volatile-Suspended-Solids (R-VSS), *etc.*

4.3 Data Gathering Method

The *Data Collection system* receives data from the on-line sensors of the plant only when these values are needed for the inference process and the values stored in the Data Base are no more valid. All the variables have a validity time interval for their values. Once this interval has been overcome, the stored value of a variable is useless, and a new value has to be acquired by means of the Data Acquisition module and the GSI (G2 Standard Interface) and has to be sent to the Data Base through the Data Base management system. The *Data Base Management system* controls the access of the several KBS, CBRL agent, Supervisory–KBS to the evolutionary (real-time) Data Base to guarantee the consistency of the system. This control is achieved by means the G2 shell mechanisms.

The required data for the deductive processes that cannot be obtained on-line, and also cannot be inferred or computed by DAI-DEPUR are required to be provided by the operator (off-line data).

Chapter 5

The Knowledge/Expertise Level

Knowledge itself is power .

Sir Francis Bacon

Studying the past, one learns for the future.

Japanese proverb

5.1 Introduction

At the knowledge/expertise level, DAI-DEPUR presents an approach to *integration* of three kinds of knowledge to model the domain: numerical control knowledge provided by an automatic control algorithm, general knowledge coming from experts' domain knowledge or expert knowledge, and specific knowledge supplied from previous solved problems in the system or experiential/practical knowledge. The numerical control knowledge is modelled by a dissolved oxygen (DO) control algorithm developed. The expert knowledge is modelled through explicit inference rules, while the experiential knowledge is modelled by means of cases or experiences. This cooperation tries to get benefit from the advantages of all kinds of available knowledge, and to cope with typical shortcomings either from knowledge-based systems: do not learn from experience, the knowledge acquisition problem, the brittleness, lack of reusability; or from automatic control systems: complexity of the processes, ill-structured domains, non-numerical or qualitative information, uncertainty or approximate knowledge. This combination of paradigms, at this level, integrates in the same architecture different kinds of knowledge and some cognitive processes as knowledge-based reasoning, case-based reasoning, learning, knowledge acquisition, problem solving [Plaza *et al.* , 1993; Gil, 1991; Newell, 1990; Van Lehn, 1990].

The paradigms of numerical control and expert systems capture the basic general principles of WWTP operation when *normal situations* and *abnormal usual situations* such as *bulking* , *storm* , *rising* , *etc* . are occurring in the WWTP, through automatic control algorithms and rule-based reasoning. They are useful in a wide variety of plants with similar technology and characteristics such as design conditions, inflow quantity, inflow quality, *etc*., but they are not able to model the particular features and specific situations (*abnormal unusual situations*) that could occur in a certain plant. Furthermore, they do not learn from the past situations occurred a given plant.

Every plant's behaviour is potentially different to other similar plants because of the changes in the inflow, meteorology, neighbouring industries, *etc*. Thus, a reliable supervisory system needs to integrate a dynamic component to adapt itself to the special characteristics of a concrete plant under control, trying to cope with the local *abnormal unusual situations*. On the other hand, to learn from previous successful situations or failed ones, it is a key fact in order to improve the performance of the supervisory system [López, 1993; Schank and Slade, 1991; Aamodt, 1989; Van de Velde, 1987b] and also, it is the same way in which human expert management works.

5.2 Domain Models

In the next subsections, the three models will be detailed. The numerical control knowledge (*mathematical algorithmic model*), the expert knowledge (*functional cause-effect model*), and the experiential knowledge (*associative experiential model*).

5.2.1 Numerical Control Knowledge

Activated sludge is, undoubtedly, the most widely extended wastewater treatment practice. In this process, a mixture of several microorganisms transform the biodegradable pollutants used as substrate or Chemical Oxygen Demand (COD), into new biomass, with the addition of dissolved oxygen supplied by any aeration system.

Like other biotechnological processes, its real-time control constitutes a quite complex problem as it has been outlined in previous chapters, due to the lack of reliable on-line instrumentation and simplicity of the models used to describe the microbiological processes that take place in the bioreactor [Beck, 1986]. In addition, it presents some specific problems like the great variability of the input both in quantity and quality and the very complex interactions between the different microorganism populations present in the system.

Effective operation can be achieved, however, by regulation of substrate levels and the maintenance of dissolved oxygen (DO) in the process above minimum acceptable conditions. On the other hand, it is important to realize that activated sludge process efficiency is independent of DO concentration, over a critical level, which is between 0.5 and 3.0 mg/L, depending on process conditions [Stenstrom and Podunska, 1980]. Then if we are able to ensure this critical DO level, it is feasible to decompose the control problem into separate multi-input single output subsystems. In this case, substrate control and dissolved oxygen control in the reactor, can be independently handled and thus, more effectively regulated.

Regulation of DO may improve the plant performance, not only from an energy point of view, i.e. by saving energy costs up to 30%, but avoiding incidences which can cause *filamentous sludge bulking* , or *poor sludge settling conditions* [WPCF, 1988].

However, the main practical problem encountered in DO control is that, the great majority of operating WWTPs do not have any provision for varying the air flow rate continuously, and the aeration equipment is operated on an on/off basis [Marsilli-Libeli, 1989]. It is the sequence of motor on/off switches which must be determined in order to control the dissolved oxygen level. Furthermore, several constraints must be satisfied in order to avoid technical problems.

Fig. 5.1. Biological balance

A non-linear predictive control algorithm has been developed to satisfy quality constraints whilst reducing energy demands [Serra *et al.* , 1993; Moreno *et al.*, 1992]. The algorithm predicts the series of motor switches in order to maintain the DO concentration always over the critical level (2 mg/L in Manresa's WWTP), which allows the normal evolution of the substrate oxidation reactions (see figure 5.1).

Fig. 5.2. Numerical control algorithm

A scheme of the non-linear predictive algorithm is depicted in figure 5.2, where the symbols and abbreviations used are:

CH : Control horizon

$\underline{DO}$: Array of dissolved oxygen values, [DO(t) DO(t+CH-1)]

F_C : Cost function value

Q : Input flow

$\underline{A}^{*}$: Array of optimal aeration factor values

$\underline{A}^{*'}$: Array of suboptimal aeration factor values, keeping the technical motor constraints.

The implementation of the dissolved oxygen (DO) control scheme is based on four main blocks:

The mathematical model of the process

A software sensor to estimate the oxygen uptake rate (OUR)

A continuous-range optimization procedure

An algorithm that using the continuous-range optimal control value computed by the previous block, generates a discrete-range suboptimal control value, suited to be applied by the aeration motors.

Mathematical model

The dynamics of dissolved oxygen in the bioreactor is described by a differential equation that has the form:

$$dC/dt = K_{la} * F(t) * (C_S - C) - OUR + f * (C_0 - C) \quad [1]$$

where

K_{la} : the global mass transfer coefficient of oxygen

C : the concentration of DO in the bioreactor (mg/L)

F(t) : the number of aerators switched on, at each moment

C_S : is the saturation level for the dissolved oxygen, and

$f * (C_0 - C)$: corresponds to the input and output of dissolved oxygen with the flow

OUR estimation

OUR is estimated using a recursive least square (RLS) method, but applying Bierman U-D factorization of covariance matrix and directional forgetting factors to improve the tracking capability of the estimation algorithm [De Prada *et al.*, 1991]. The identification algorithm uses a discrete model of the form:

$$DO(t) = f(t) * q^{T}(t)$$

where

$$q(t) = [K_{la}, OUR]$$

and the measures of the array f (t) are given by a linearization and discretization of equation [1].

Continuous range optimization algorithm

Continuous optimization is performed using a modification of the Broyden-Fletcher-Shanno algorithm, to take into account the fact that power aeration is constrained between zero and an upper value. The modification employed uses a projection procedure in order to translate, into the feasible space, the search points located out of it [Moreno, 1991].

A sequence of control actions (on/off motor switches) is defined for a prediction horizon (PH). A continuous non-linear mathematical model of the DO dynamics is simulated by the ACSL continuous simulation language [Mitchell & Gautier, 1987], using equation [1], in order to predict the DO evolution through this period, and give enough information to compute the value of the cost function F_C.

Generation of aeration motor switches

Indeed, from the six tanks in which the activated sludge reactor is divided in Manresa's WWTP, the first two have motors with two different speeds, corresponding respectively to 60 and 90 hp, meanwhile the other four work only at fixed speed, corresponding to 75 hp. Thus, the number of different values for the aeration factor at each sampling is given by $3^2 * 2^4 = 144$. There exist some equivalences between all that possibilities due to symmetric considerations. Taking into account these considerations, the different values of the aeration factor can be reduced from 144 to 27.

Anyway, with 27 different action possibilities, considering a control horizon of 3 sampling periods, as in our case, this involves $27^3 = 19683$ different combinations to evaluate at each sampling instant. Under this situation, trying to solve the optimization problem from a discrete point of view, seems to be unapproachable. Thus, the adopted solution was to decompose the optimization problem into two subproblems: a continuous one, where an array of optimal values $\underline{A}^* = [a^*(t)\ a^*(t+1)\ a^*(t+CH-1)]$ is obtained; and a second one, in which, starting from time t, to time t+CH-1, feasible combinations of aeration motors switches are evaluated, giving values of $\underline{A}^{*'} = [a^{*'}(t)\ a^{*'}(t+1)\ a^{*'}(t+CH-1)]$ as near as possible to the optimal $\underline{A}^*$ array, and keeping the technical constraints of the Manresa's WWTP, which are:

All motors have to keep the selected speed, as a minimum, for one hour. This is a mechanical constraint required to avoid an excessive wear and power consumption of the motors.

No motor can be switched off for more than one hour. This is a biological constraint required to avoid the sedimentation of biomass into the reactor.

Also the *Numerical Control Knowledge module* allows DAI-DEPUR to simulate the WWTP operation, to obtain simulated values from some required variables, *etc* .

5.2.2 Expert Knowledge

In this paradigm, the expert knowledge about the domain is modelled with inference rules. They form the different knowledge bases of the distributed Knowledge-Based Systems integrated in DAI-DEPUR.

5.2.2.1 Inference Rules

One of the inherent difficulties to the development of a knowledge-based system (see figure 5.3) is building-up the knowledge base (i.e., *knowledge acquisition*), specially when dealing with a wide and complicated field (i.e., *ill-structured domain*) [Steels, 1990; Becker, 1987].

Acquiring relevant knowledge is a difficult task for a number of reasons. Experts do not necessarily find it easy to formulate how to solve the day-to-day problems arising in the running of the process. Moreover, they may omit part of the information in their explanation, introduce inconsistencies, or be unable to explain which sequence they should follow in order to solve a problem. In addition, they may not be very familiarized with the knowledge representation structures (inference rules, *etc* .) used in the knowledge-based systems to embody the knowledge in a computational model. Furthermore, *Situations* occurring in the WWTP are often simultaneously both cause and effect, or are closely interrelated, thus obscuring the causes of what is actually occurring.

Fig. 5.3. Design process of a knowledge-based system

All these problems make it difficult for the engineer compiling the knowledge base to determine the relevant parameters, without introducing errors. He may also, mistakenly, consider some information to be irrelevant or redundant, or alternatively, make things unnecessarily complicated.

The process of extracting these rules (*knowledge acquisition*) from experts has been traditionally done with *interactive sessions* between the experts and the knowledge engineer, where the experts try to make explicit their knowledge and reasoning processes. From that knowledge, the knowledge engineer tries to structure it and he/her gets what has been called as a *decision tree* (see figure 5.4).

Fig. 5.4. Example of a decision tree

A decision tree is a set of nodes and arcs. Each node is a question related to a concrete information (for example, the value of an attribute, *etc*.), and each of its nodes is a possible value for that information. The leaves of the tree correspond to a classified objects (for example *situations* , causes, *etc* .). From the structured information of the decision tree, inference rules can easily be formulated. But the manual construction of decision trees is quite often described as a bottleneck. Therefore, much effort in AI has been addressed to overcome it.

Some techniques have been developed to *expert interactive knowledge explicitation* [Booser and Bredshaw, 1987; Plaza, 1987; Clancey, 1985a]. They are tools and/or models based on specific tasks such as explicitation, acquisition, and concept refinement. They are easily applied to several domains, but they are not fully automated.

The main features of the domains in which knowledge acquisition is required, suggest the idea to use some *machine learning methods* to model the knowledge domain. Thus, the task of the experts is reduced to the work of providing the machine learning techniques with good *examples* and their validation. An alternative to conventional machine learning systems of determining parameters, based on *supervised learning of discriminating or structural descriptions* [López de Màntaras, 1991; Kononenko and Bratko, 1991; Danyluk, 1987; Quinlan, 1986; Michalsky and Reinke, 1986; Mitchell, 1982; Dietterich and Michalsky, 1981; Vere, 1980; Hayes-Roth and McDermott, 1977; Winston, 1975], is the use of *techniques based on unsupervised learning of taxonomic descriptions* [Fisher and Pazzani, 1991; Dubes and Jain, 1990; Gennari *et al.* , 1989; Cheeseman *et al.*, 1988; Lebowitz, 1987; Hanson and Bauer, 1986; Lenat, 1984; Michalsky and Stepp, 1984], more suitable – on the other hand– to be applied to *ill-structured domains* (see [Ke and Ali, 1991] for an overview and extensive bibliography on inductive learning methods). When an expert rationalizes about his domain, he organizes his observations and groups them into categories using some similarity criterion. At a more advanced cognitive level, he groups these categories in order to create general criteria which will enable him (or others) to discriminate among objects (or events) and make decisions. The advantages of organizing information into categories are: categories diminish the complexity of the domain by simplifying the problem; they relate observations; they act as the first step in the identification process and make it possible to predict the characteristics of a new observation.

Moreover, from these categories or concepts (*situations*) discovered, a set of inference rules (KB) can be generated, that lead to identification and diagnosis of the current operating situation of the plant.

In our case, the knowledge acquisition process have been done using LINNEO $^{+}$ (a semi-automated unsupervised classification tool, that it is described in 5.4.1.1) [Béjar, 1995; Béjar *et al.* , 1994; Béjar and Cortés 1992]. The main objective of LINNEO $^{+}$ is to build classifications for ill-structured domains; where much imprecise information exists, it is assumed that observations vary in their degree of membership with regard to each class. Bearing all this in mind, LINNEO $^{+}$ uses the conventional concept of distance as a fuzzy similarity value.

The results of the classification process using LINNEO $^{+}$ provides intensive and extensive description of the generated classes, *situations* in this domain, and a *fuzzy membership matrix* that relates observations (data) to generated classes. At this point, inference rules can be generated, both manually or using GAR [Riaño, 1994]. Bearing in mind the *prototype* of a class and the superclassification structure, inference rules that lead the *diagnosis* process in the target system, can be derived. These rules *identify* to which class or set of situations a given observation (data) belongs.

Identifying rules characterize the values that descriptors of new data must show for being member of a class. For instance a rule as the following could be generated:

(if (COD-EXIT HIGH

SVI HIGH

SS-RECIRCULATION LOW

SSV-RECIRCULATION LOW)

VERY-POSSIBLE

(INFER CLASS-3A))

Other *discriminating rules* must be provided to discriminate to which *situation*– within a class– belongs a given observation. For this task is taken into account the *fuzzy membership matrix* . For example:

(if (CLASS-3A

SLUDGE OLD

FILAMENTOUS NORMAL)

ALMOST-SURE

(INFER BULKING-NON-FILAMENTOUS))

This process is semi-automatic, because it requires the experts' final validation. All these rules are analyzed using subsumption detection, synonymy analysis, *etc*. and, afterwards, they are organized in a hierarchical way to guide the diagnosis process. The results are made known to the experts who can accept and confirm them, or have the chance to go back to the classification process. When all these actions are over, rules can be validated using new observations not previously in the classification sample. As soon as rules are accepted by the experts, they can be incorporated into one of the KBS of the distributed architecture for a later Rule-based Reasoning method (see figure 5.5). These rules capture the subjective domain knowledge of the experts in their daily work at the WWTP. The coordination among the various KBS allows the diagnosis process that leads to identify the *generic working situation(s)* of the plant.

Fig. 5.5. Expert general knowledge

If there exists some disagreement among the experts opinions or points of view, it is possible to use EGAC [Torra and Cortés, 1995], to try to convey a consensus among their judgements.

5.2.2.2 Distributed agents' knowledge

All knowledge and information involved in the operation of a WWTP is spatially distributed among different operational units located at different sites of the WWTP. Thus, taking into account the two major subsystems of an activated sludge WWTP, we can describe each one of the Knowledge-Based Systems (agents) in which the knowledge has been distributed achieving certain level of modularity and independence. The agents forming the Water line subsystem are (as shown in figure 5.6):

Screen– KBS: Agent supervising physical units that remove gross pollutants from the inflow of the WWTP. It is the first physical unit of the WWTP, and is responsible to remove the big solids present in the inflow water, such as plastics, branches, bottles, packs, etc. It takes care of the cleaning automatic grids, that usually scan the water at fixed intervals of time. At the same time, it must control the transportation ribbons who move the captured solids to a container. It is one of the analytical sample point in the WWTP (I-COD, I-BOD_5, I-SS, I-VSS, I-pH, *etc*.)

Grit removal– KBS: Agent supervising physical units that remove grit to prevent abrasion and wear of mechanical equipment. At the same time, with a suplementary providing of air, the grease is sent to the water surface, which are guided to a static grease separator and discharged to another container. It controls the correct operation of the mechanical objects involved. The on-line value of the inflow-water is measured in this unit through a Parshall channel, and some off-line qualitative information (floating accumulation, *etc* .) is acquired.

Fig. 5.6. Water line KBS agents

Primary settler– KBS: Agent supervising physical units with a long residence time that remove suspended particulates heavier than water. These solids are purged at the bottom of the primary settlers and sent to the sludge line for a later treatment. It must control the rotatory bridges of the primary settlers that pushes the sludge to go out, and the waste flow of the primary settlers. Also, some off-line qualitative information (presence of bubbles, *etc* .) is acquired within it.

Biological reactors– KBS: Agent supervising biological units that, using aerobic microorganisms (biomass or activated sludge), convert soluble BOD to new microorganisms. This biological units are the key of the activated sludge WWTP operation. These units acquire the DO on-line measure and contain another off-line analytical sample point (SRT, SVI, B-SS, B-VSS, *etc* .), and some off-line qualitative information (microscopical information, water odour, water colour, *etc* .).

Secondary settler– KBS: Agent supervising physical units with a long residence time that separate biomass from the liquid phase. A correct separation process is crucial for the WWTP operation, if there is not a tertiary treatment. It must control the operation of the rotatory bridges of the secondary settlers (clarifiers). Another analytical sample point in these units acquire off-line analytical information (O-SS, O-VSS, O-COD, O-BOD_5, *etc* .), and some off-line qualitative information (presence of bubbles, *etc* .).

Chlorination– KBS: Agent supervising chemical units that allow, if necessary, to desinfectate the outflow prior to discharge. Nowadays, it is not advised to normally use it, but it can be used to remove the pathogen microorganisms from the effluent water. The chlorination channel must be periodically cleaned from sludge that escapes from the secondary settlers.

Recirculation–KBS: Agent supervising pumping systems to keep a certain level of biomass in the biological reactors. In this unit it is measured some analytical off-line information (R-SS, R-VSS). The pumping system uses to be an Archimedes' screw.

The agents forming the Sludge line subsystem are (as shown in figure 5.7):

Waste– KBS: Agent supervising pumping systems that determine the sludge age (Mean Cell Residence Time, MCRT).

Thickment– KBS: Agent supervising physical units to increase the sludge concentration.

Anaerobic treatment–KBS: Agent supervising biological units that, using anaerobic microorganisms, convert biomass in methane (biogas). Its goal is to decrease the percentage of BOD about 80%.

Drying– KBS: Agent supervising physical units that dry the sludge prior to discharge.

Fig. 5.7. Sludge line KBS agents

5.2.3 Experiential Knowledge

We propose to use Case-based Reasoning (CBR) to model the experiential specific knowledge about a concrete WWTP. CBR is a flexible paradigm that supports the implementation of a dynamic learning environment. Within the frame of a Case-based reasoning agent, we can model the actual operating situations of a WWTP through cases, and organize all the cases into the case library.

CBR systems have been used in a broad range of domains to capture and organize past experience and to learn how to solve new situations from previous past solutions. CBR systems have been applied to planning (CHEF [Hammond, 1989]), design (JULIA [Hinrichs, 1992]), classification (PROTOS [Bareiss, 1989]), diagnosis (CASEY [Koton, 1989]), understanding and analysis (AQUA [Ram, 1993; Ram and Hunter, 1992]), interpretation (HYPO [Ashley, 1990]), and explanation (SWALE [Kass and Leake, 1988]). For Case-based Reasoning in continuous situations, we have only knowledge about the system NETTRAC [Brandau *et al.*, 1991] as a Case-based system for planning and execution monitoring in traffic management in public telephone networks. In the WWTP domain, Case-based Reasoning has been used for designing most suitable WWTP operations for a set of determined input contaminants [Krovvidy and Wee, 1993].

Case-Based Reasoning [Aamodt and Plaza, 1994; Kolodner, 1993; Riesbeck and Schank, 1989] derives from a view of understanding problem solving as an explanation process. The foundations or origins of Case-based Reasoning rely on the early work done by Schank and Abelson [Schank and Abelson, 1977] where they proposed that our general knowledge about situations is recorded as *scripts* . The cognitive model behind the Case-based reasoning is based on the theory of *Dynamic Memory* [Schank, 1982] that introduces indexing as the key to use experience in understanding. The main premise was that remembering, understanding, experiencing, and learning cannot be separated from each other, and that the human memory is dynamic, and change as a result of its experiences.

Fig. 5.8. The Case-based Reasoning paradigm

In our approach, the knowledge about the practical problem solving – wastewater treatment plants operation– in the domain is represented by means of cases or experiences that include the description of *situations* , which are organized in the case library. This case library contains information about previously detected *situations* and the *solutions* given to them as well as a measure of their *efficiency* (specific experiential knowledge).

CBR systems improve their performance becoming *more efficient* by remembering old solutions given to similar problems and adapting them to fit a new problem rather than having to solve it from scratch . This, in fact, augments the ideas about the components of expertise [Steels, 1990] using the solved cases as an episodic memory: the memorization of problem-solved episodes allows methods to be integrated since they require to access the past experience to improve the system performance. Also, Case-based reasoners become *more competent* in their evolution over time, so that they can derive better solutions when faced against less experienced situations, preventing them to repeat the same mistakes in the future (learning process).

The reasoning and learning processes in a Case-based system are performed by the following steps (see figure 5.8):

Retrieving the most similar case(s) or previous working situations to the new case, by means of a recalling algorithm based on partial-matching and some heuristic functions or distances, possibly domain dependent, to select the best case(s).

Adapting or *reusing* the information and knowledge in that case to solve the new case, i.e. the current working situation of the plant. The selected best case has to be *adapted* when it does not match perfectly the new case. Adapting methods can be used to insert something, or delete something, or make a substitution in the selected case to solve the new one. In systems as CABINS [Miyashita and Sycara, 1995] there are included strategies as knowledge filtering to validate the effectiveness of selected adaptation actions, and to give-up further adaptation if the likelihood of success is low.

Evaluation of the proposed solution. A Case-based reasoner must require some feedback to know what is going right and what is going wrong. Usually, it is performed by simulation or by asking to a human oracle . In the future, the evaluation will be done through automatic checking of the effectiveness of past solved cases.

Learning the parts of this experience likely to be useful for future problem solving. The agent can learn both from successful solutions and from failed ones (repair). This goal can be achieved either by updating the Case Library accordingly: adding cases (learning), deleting cases (forgetting), modifying cases, changing indexes to cases, global reorganization of the case library, *etc* . or changing the distance measure [Wess and Globig, 1994].

As in other systems, like SOAR [Newell, 1990], there are two kinds of impasses: either the situation is unknown, i.e. there is no memory about this situation or there is not a successful solution to this situation, or there are several ways (solutions) to proceed i.e. there are several methods that may be applicable to a situation with the same degree of confidence. For these impasse situations, DAI-DEPUR architecture can use the expert general knowledge coded into the system or can generate an alarm that has to be solved by the operator. Other approaches as NOOS [Arcos and Plaza, 1995] generate a reflexive task whose goal is to solve that impasse.

5.2.3.1 Missing information

Case-based reasoning is also useful when knowledge is incomplete or uncertain, even using an attribute-value representation. A Case-based reasoner can make assumptions to fill-in incomplete or missing knowledge, based on its experience and can continue reasoning from there.

Our Case-based reasoner accepts that a possible value for an attribute is an *unknown* or *nil* value. A *nil* value could be a missing or unknown or do not care (nought) value. Many times, the values coming from some sensors in the plant or provided to the system by the operator of a plant are wrong and sometimes, some values are just missing, and the system has to be performant enough to maintain the operation.

Missing information is the first main problem within discrimination trees or networks organization. When a new case is incomplete, there is no guidance about how to continue searching when a test, in the discrimination tree, cannot be answered. There are four options to solve this problem: stop the search, continue searching on all possible childs (paths) of the node, start the retrieval in another discrimination tree organized with a different order of the tests on the attribute's values, i.e. redundant discrimination trees, and choosing the most plausible branch.

All the options are problematic [Porter *et al.* , 1990]: stopping the search is an unsatisfying choice that can cause an impasse to the Case-based reasoner; searching on all possible paths is an expensive alternative that it would worsen the retrieval time; using redundant discrimination trees causes an overhead in time and space, although it has been used in MEDIATOR [Kolodner and

Simpson, 1989], CYRUS [Kolodner, 1985] and CASEY [Koton, 1989]; the most promising branch is perhaps the most satisfying solution, although it requires additional computation to look ahead in the tree or the addition of some knowledge that can be used to make the choice without looking ahead.

In our Case-based reasoner, we have implemented a technique to choose the most promising branch. The method tries to get benefit from the dynamic learning environment. There is a *frequency value* associated to each arc of a node of the discrimination tree, which is the data structure that implements the Case Library (as explained in 5.2.3.4). This frequency value measures the number of times that this arc has been traversed when retrieving cases with the same values on the previous attributes of the tree, i.e. the same partial-matching cases. In fact, it is a weighted measure of the traversals of the arc depending on the type of exploration. So, this is a *dynamic measure* that is being adapted as the case library evolves over time. When there is a missing value for an attribute, then the branch (arc) with higher frequency value is selected. In fact, as is explained in 5.4.2.2, the two arcs with highest frequencies are used.

5.2.3.2 The table of attributes

An important component for the system's behaviour is the construction of the table of attributes. The *operating situations* are described by means of the attributes of this table. The situations are described with 11 attributes selected among the 38 that were defined by the experts to describe the domain. All these features are stored in an attribute's table as showed in table 5.1.

All the quantitative or lineal (ordered) attributes were discretized by the experts into several modalities, commonly, between two and four modalities, when they were defined to the system. The qualitative attributes (not ordered) are already differentiated into several categories at the definition stage. An outstanding feature of the system is that the definition of the table of attributes is *independent* from the general Case-based reasoner operation. Each attribute has a weight measure (between 0 and 10), also defined by the experts, that shows the attribute's relevance in the characterization of a situation.

ATTRIBUTE	INTERPRETATION	WEIGHT	MODALITIES
I-pH	pH at the input of the plant	5	Acid, Neutral, Basic
I-SS	Suspended solids at the input	6	Low, Normal, High
I-BOD	Biodegradable organic matter at the input	6	Low, Normal, High
I-COD	Chemical oxidable organic matter at the input	8	Low, Normal, High
I-Zn	Concentration of zinc at the input	5	Normal, High
Q	Inflow wastewater	7	Low, Normal, High
D-SS	Suspended solids at the end of the primary treatment	4	Low, Normal, High
D-COD	Chemical oxidable organic matter at the end of the primary treatment	4	Low, Normal, High
O-SS	Suspended solids at the output	9	Low, Normal,

			High
O-COD	Chemical oxidable organic matter at the output	9	Low, Normal, High
O-pH	pH at the output of the plant	5	Acid, Neutral, Basic

Table 5.1. The table of the attributes defined in the Case-based agent

5.2.3.3 Cases

The cases stored in the case library are real WWTP operating experiences, which have been captured and learned in such a way that they can be reused to solve future causalities. A case does incorporate a set of features such as an *identifier* of the case ; the *description of the situation* ; the possible *diagnostic* of the situation; the possible *action plan* ; the *derivation* of the case (from where the case has been derived/adapted); the *solution result* , information indicating whether the proposed case solution has been a successful one or not; an *utility measure* of the case in the system; the *distance* value, it is a measure of similarity of the case when it was retrieved last time from the case library. An example of a case representation is:

(:identifier CASE-18

:situation-description ((Q 35,198 m^3/day)

(I-COD 289 mg/L)

. . .)

:diagnostics NORMAL-SITUATION

:action-plan ((1 Maintain-the-numerical-control-algorithm)

(2 Adjust-Dissolved Oxygen (DO)-value)

. . .)

:case-derivation INITIAL-CASE / CASE-13

:solution-result SUCCESS /FAILURE

:utility-measure 0.7

:distance-to-case 0.3782)

5.2.3.4 The case library

The case library is organized in a hierarchical way improving the time access to the stored cases in the retrieval phase. It is implemented as a *prioritized discrimination tree* [Kolodner, 1993; Charniak *et al.*, 1987]. Each non-terminal node of the tree is a test on the value of the attributes. The priority ordering of the attributes has been obtained from the experts' opinion and has also been validated against an inductive machine learning algorithm (ID3 [Quinlan, 1986]) with very slight differences in the ordering [Segarra, 1995]. The prioritized discriminant list of attributes is:

(I-Zn O-SS O-COD I-SS I-COD I-BOD Q D-SS D-COD I-pH O-pH)

Obviously, the retrieval time of cases depends on the power of the discriminating order of the attributes stated by the experts. If the discriminating order is good, then it will be only needed to make a few tests on the values of the attributes. But if the ordering is bad, then too much tests will be needed to retrieve the cases.

Each branch of a node (attribute) is a possible qualitative value for the attribute. The terminal nodes (leaves of the tree) have the recorded cases that suit all values in the branches, from the root to the leaves. An example of a Case Library generated with the system is depicted in figure 5.9.

Fig. 5.9. A Case Library example

5.3 Tasks

The two main tasks involved in the knowledge/expertise level are the *diagnosis* task which discovers what is happening in the WWTP operation, and the *adaptation* task which enables DAI-DEPUR to not degrade its performance over time.

5.3.1 Diagnosis

Diagnosis is the task responsible to identify the current working situation of the WWTP and the possible causes that lead the WWTP to that *abnormal situation*, if it is the case. The task is achieved from the evaluated data gathered at the previous level.

Fig. 5.10. Diagnosis process in a knowledge-based system

The diagnosis process is made through two steps (as it is depicted in figure 5.10). At first step, it is intended to determine the working situation of the WWTP (*i.e.*, the problem identification), and if the diagnosed situation is *abnormal* , there is a second stage which is intended to discover the possible causes that originated that anomalous situation (*i.e.*, the cause identification).

The diagnosis task is implemented in a complementary twofold way: through a *Rule-based reasoning* method, that makes use of the distributed expert knowledge encoded into DAI-DEPUR in form of rules, and by means of a *Case-based reasoning* method, that retrieves the most similar experienced situation on that WWTP (case) that has been recorded in the Case Library. This multi-paradigm integration tries to get benefit from both methods advantages and to cope with a possible impasse generated from one approach, when it is not able to diagnose anything at all. Another impasse can be originated when the two diagnostics are contradictory, which may be solved at the situations level, by the WWTP operator's supervision.

5.3.2 Adaptation

The *adaptation* task has the primary goal of providing DAI-DEPUR with a dynamic adaptive behaviour. In order to solve the brittleness of most AI systems when faced against real-world domains, DAI-DEPUR incorporates the adaptation task which is performed by means of the acquisition of new expert knowledge from (new) experts/sources (*expert knowledge acquisition* method), through the acquisition of new observed experiential knowledge (*learning from observation* method), by means of learning from new experiences (*learning from experience* method), and finally, through the Case Library updating (*introspection* method).

This adaptation capability enables DAI-DEPUR to be reused to supervise different WWTPs with the same or slightly different technology, so that the adaptation methods will allow it to adapt to the new characteristics of the new WWTPs under supervision.

5.4 Methods

The methods that perform the diagnosis and adaptation tasks will be detailed in the next sections: *expert knowledge acquisition* and *Rule-based Reasoning* methods work on the expert knowledge, whereas *learning from observation*, *Case-based reasoning*, *learning from experience* and *introspection* methods work on the experiential knowledge.

5.4.1 Expert Knowledge Methods

5.4.1.1 Expert Knowledge Acquisition

The *Expert Knowledge Acquisition module* is based in recent developments in knowledge acquisition. This module uses the software μ , which is the merging of LINNEO$^+$ ([Béjar, 1995; Béjar *et al.*, 1994; Béjar and Cortés, 1992] and GAR [Riaño, 1994] for automatic generation of inference rules as the result of a previous classification process of attributes and observations, defined by experts ([Sànchez *et al.*, 1995e; Serra, 1993]).

Fig. 5.11. LINNEO$^+$ methodology

LINNEO$^+$ is a knowledge acquisition tool that *incrementally* works with an unsupervised learning strategy which accepts a stream of observations and discovers a classification scheme on the data stream. As a control strategy, it retains only the best hypotheses which are consistent with the observation given a similarity criterion. Part of the LINNEO$^+$ methodology (see figure 5.11) could be considered as conceptual clustering with two critically important tasks:

Clustering, which determines useful subsets of data using a fuzzy set approach, and characterization, which determines a concept for each extensionally defined set discovered by clustering.

The second task requires external help (*validation* from experts) to accept or reject the resulting clusters. Other modules try to exploit observational knowledge from the data set, or take advantage of the experts' knowledge if available. This knowledge is called Domain Theory (DT), and its use to semantically bias the process is fully explained in [Béjar *et al.* , 1994].

The main objective of LINNEO^{+} is to build-up classifications for ill-structured domains; where much imprecise information exists, it is assumed that observations vary in their membership degree with regard to each possible class in the domain. Bearing all this in mind, the use of the conventional concept of distance as the complementary function of a fuzzy similarity is used. Objects are represented as vectors of length n , n being the number of descriptors. Position k of object O_i shows the symbolic or numerical value of descriptor (O_{ik}). The distance which will be used for determining the similarity between two objects, O_i and O_j, is the generalized Hamming distance:

$$\mathrm{dist}(O_i, O_j) = \sum_{k=1}^{n} \mathrm{diff}(O_{ik}, O_{jk})$$

where

$$\text{diff}(O_{ik}, O_{jk}) = 1 \text{ if } O_{ik} O_{jk}$$

$$\text{diff}(O_{ik}, O_{jk}) = 0 \text{ if } O_{ik} = O_{jk}$$

when k is a qualitative attribute, and

$$\text{diff}(O_{ik}, O_{jk}) = |O_{ik} - O_{jk}|$$

when k is a quantitative attribute.

The centre of a class is obtained by calculating the mean value for each quantitative attribute of every object. For qualitative attributes, the centre includes each of its modalities with its corresponding occurrence frequency. Note that the centre of a class can be considered as the *prototype* of the objects contained in the class. The distance between an object and a class prototype can be taken as the complementary function of the degree of fuzzy membership of the object O_i to the class C_j:

$$\text{membership}(O_i, C_j) = 1 - \text{dist}(O_i, \text{centre}(C_j))$$

Note that the distance is always computed between a new object to be inserted within the objects' space and a *prototype of a class* (excepting the time when a class is formed of only one object), which has a frequency distribution of the values of the qualitative attributes and the mean value of the values for the quantitative attributes. The distance is computed as:

1 – relative frequency of qualit. value, for the qualitative attribute's value

|mean prototype value – value|, for the quantitative attribute's value

Therefore, there is not a binary (0/1) measure as seems to be by the formalism above proposed. If a value for the distance is established as the limit for an object to belong to a class (a limit which is called *radius*), then a classification rule (expressing the similarity criterion) is found:

IF $\text{dist}(\text{centre}(C_i), O_j) < \text{radius}(C_i)$

THEN O_j can belong to C_i

ELSE O_j cannot belong to C_i

The radius could be determined empirically in a random way if desired, but LINNEO$^+$ offers a previous data analysis facility. This is a scanning algorithm that starting from an initial radius, given a determined radius step, and ending to a top radius (all three values specified by the expert), it analyzes the number of objects that would belong in the class that has its centre at each object of the domain and within the concrete radius. Then, it computes the mean value of this number for all the objects of the domain, with the same radius. All this process is repeated with each one of the possible values of the radius. So, after this analysis, it is possible to find a value for the radius, so that, when there is not a continuous increase in the mean number of objects, it is certainly possible that the "optimal radius" will be in that interval.

In this context, the radius (R), which in this version of LINNEO$^+$ is always constant and the same for every class, can be interpreted as the degree of selectivity of the classification. LINNEO$^+$ is a classification methodology which uses analytical and empirical techniques for knowledge acquisition.

This methodology works by defining a space of *n* dimensions, in which *n* is the number of attributes. Within this space, each class is specified by a centre and a radius. All those objects which, occurring within that space, are at a given distance from the centre that is less than the radius, form part of that class. At the beginning of the process, the classes are still undefined, thus the first object is taken and placed within the space to form a class. Its centre is the point at which the object is situated, according to the value of the attributes defining it. Then, the second object is taken, and it too is placed within the space. If it is in the first class, its centre is re-calculated using the mean of the two objects. Otherwise, a new class is formed. The centre of the classes will continue to change throughout the process, depending on the objects which are included. It may happen that an object remains outside the class because the centre of that class is displaced. Finally, a set of classes with a number of objects is obtained. The class is specified by the final centre and radius of the class. A more detailed description of how LINNEO $^+$ works, together with a description of its potential, is given in [Béjar, 1995; Béjar *et al.* , 1994].

GAR (Automatic Rule Generator) is used to generate a set of classification rules from LINNEO $^+$'s output (a representation of the concept structure of the domain in terms of classes). GAR can generate both conjunctive and disjunctive rules, but after having analyzed and compared several kinds of classification rules, one arrives to the following conclusions:

The effectiveness of rule generation (defined as the specificity of a rule normalized in time) for conjunctive rules is the highest one.

When delivered to experts, conjunctive rules are qualified as more understandable than other sort of rules.

When applying conjunctive rules, the reasoning process is faster.

Conjunctive rules structure knowledge in a more modular way.

Therefore, these facts drive the system to output conjunctive rules. The algorithm for conjunctive rule generation could be summarized in this way:

1. Select the best term

2. Add such term to the "up to now" conjunctive premise

3. Reduce the set of possible terms

4. Repeat steps 1 to 3 while the rule is not completed

For instance, a rule generated by GAR is:

((> 323.0 D-COD)

(< 7.7 I-pH)

(< 93.0 GENERAL-CLEANSING-PERCENTAGE-BOD)

(> 300 I-BOD $_5$))

– >

Class-12)

which describes a *high in-plant-overloading situation* plus a poor sedimentation process in the primary settler. Although some of the automated rules generated by GAR, can be easily interpreted by the experts, they feel that some of those rules are very specific, in the sense that they can only be applied in few cases. An abstraction task and other strategies are now being developed in the automatic rule generator.

5.4.1.2 Rule-based Reasoning

Rule-based Reasoning is one of the methods that allows DAI-DEPUR to diagnose the current working situation of the WWTP, from local diagnosis performed by the different Knowledge-Based Systems. Rule-based reasoning is carried out by the inference engine provided by the G2 shell. This shell allows the basic kind of knowledge inference: backward and forward chaining.

The *backward chaining* is used to find a value to a variable, that cannot be give by on-line sensors, by simulation or by some formula. Also, it can be used to infer a boolean/fuzzy fact. In backward chaining, the reasoning is guided from the conclusion(s) –that is(are) wanted to deduce– to the premise data. Two strategies can be defined to implement the backward chaining: *depth-first backward chaining* and *breadth-first backward chaining* . The first one selects the new invoked rule as the next rule in depth from the possible applying rules tree generated. The second one selects the new invoked rule as the next rule in breadth from the possible applying rules tree.

Forward chaining is guided from the premise data to the conclusions of the system, by means of deducing some new facts from previous ones. It is usually applied when there are no possible hypotheses to validate, and it is desired to deduce new knowledge as much as possible.

There are other techniques that can be used in the G2 shell to control the reasoning process (*meta-reasoning*) such as organizing the rules in workspaces (modules), focusing on rules, invoking rules, prioritizing rules, *etc* .; and to *handle real-time facilities* such as scanning rules, data seeking, *etc*.; to make easier the design process of the KBS such as applying generic rules, *etc* .

Fig. 5.12. A set of diagnosis rules

From the point of view of WWTP operators, the diagnostic inference rules can be grouped in: *diagnosis rules* , *fault detection rules* and *prevention rules*, although from a logical point of view of a knowledge engineer, all of them are diagnosis rules that allow to identify certain operating states of the WWTP, through the local distributed knowledge rules, and the combination (global diagnosis) rules that will be described in chapter 6. Next, we will detail some local diagnosis rules (although the combination rules are also detailed for a better understanding of the diagnosis reasoning) by means of some examples of DAI-DEPUR performance.

Diagnosis rules

In figure 5.12, a set of rules that conclude a particular situation is shown as example of how Rule-based Reasoning carries out diagnosis. These rules allow to infer an organic overloading from in-plant *sidestreams* when it occurs. In the different Knowledge-Based Systems of DAI-DEPUR, this situation is called *sidestreams* .

In this case, not only classic diagnosis is done but also failure detection because, as COD is measured only at the input of the plant, organic loadings inside the plant (after the input sampling point and before the biological process) are not known. There are not flow sensors nor analyses done in these points to detect the situation. However, if special care were taken, estimation of COD loading would be possible.

Detection of failures

Sometimes, during the Rule-based Reasoning, the diagnosis rules do not establish the cause of a concrete problem. Under these circumstances, the failure detection rules are performed. These rules include the heuristic knowledge corresponding to the concrete situation produced when a WWTP element is malfunctioning. Rules for the detection of failures in the WWTP sensors, pipes and weirs have been defined into the different KBS. Figure 5.13 (a) and (b) shows two rules as examples of possible disturbances, as well as their solving actions.

In the first example, a failure in the pH sensor is observed taking advantage of the feature that the *fungi* only grow in acidic conditions, and of the fact that the pH-status is normal (SCREEN-KBS-RULE-005). This will allow to conclude, in the supervision task, that if *fungi* are present (BIOLOGICAL-REACTORS-KBS-RULE-053), and the measured pH values is not low, there must be a fault with the sensing element (SUPERVISORY-KBS-RULE-017).

In the second case, the rules will allow the WWTP operator, in the supervision task, to know whether the wasting flow pipe is plugged. For Manresa's WWTP, due to a design mistake, this situation is a rather common failure. If the biomass in the reactor is greater than the normal value (BIOLOGICAL-REACTORS-KBS-RULE-025), and COD removal efficiency is normal or high, then the cause must be a bad wasting schedule. But as the wasting flow is read on-line in the database, and the bad wasting situation is rejected (WASTE-KBS-RULE-001), then the cause inferred is a plugged pipe (SUPERVISORY-KBS-RULE-036).

Fig. 5.13 (a) and (b). Rules for fault detection

Prevention rules

One of the most valuable characteristics of an expert in WWTP relies on his/her ability to predict possible future *abnormal situations* when a specific current situation is occurring in the WWTP. Main causes – but not all– for many WWTPs upsets are known. However, it can take a long time (days or even weeks) from the revealing of the malfunction since the causes are present for the first time. So, it can happens that nobody detects that the plant is going to be in a problematic *abnormal situation* .

Bulking is a typical WWTP *abnormal situation* (see the chapter 6) where all the above mentioned is specially true. There are different causes that can lead to *bulking* situation. Most of these causes do not lead to this state immediately, but the effect appears with a middle-long term delay. Possible causes leading to the *bulking* situation are low pH values at the inflow, extreme low or high F/M (Food/Microorganisms) ratio, or extreme low or high Dissolved Oxygen (DO). Moreover, the appearance of these values does not lead directly to a *bulking* situation, as they are not a sufficient condition, but it is necessary to watch their evolution in order to inform that the conditions for *bulking* are being reached.

The set of prevention rules try to avoid these situations. Their aim is to detect and to prevent possible trouble. They scan the process looking for any situation that could lead to WWTP malfunction. When a possible upset is detected, DAI-DEPUR will try to conclude, in the supervision task, the variable that must be changed and what must be corrected to avoid it.

Two examples are presented in figure 5.14 (a) and (b). The first example describes that a high variability of pH values at the inflow can cause *bulking* (SCREEN-KBS-RULE-010). The Input-pH-variability is set according to experimental (historical) data (SUPERVISORY-KBS-RULE-042).

Usually, there are two channels opened at the plant entry, and a third one is closed. If there is an inflow increase (GRIT-REMOVAL-KBS-RULE-003), the third channel probably needs to be opened to avoid an overflow, or some other actions need to be taken to cope with a storm (SUPERVISORY-KBS-RULE-049). In the Manresa's WWTP case, the gate must be opened manually because it is not automated. Also, it must be taken care to avoid the plugging of the input screens, as the inflow may carry several objects and material (dead animals, vegetables, branches, plastic wastes, *etc* .).

Fig. 5.14 (a) and (b). Prevention rules

5.4.2 Experiential Knowledge Methods

5.4.2.1 Learning from Observation

It is important to remark that a Case-based reasoner starts with a representative set of cases. They are like the training set of other supervised machine learning methods. To this end, the initial Case Library was selected with some situations obtained by LINNEO $^+$ classification [Béjar, 1995], from a real data stream of 527 data (days) corresponding to the period 1990-1991. These data are available by anonymous ftp from UCI machine learning repository of data bases (ftp.ics.uci.edu). Each data is described by means of the daily mean of 38 variables. That study [Sànchez *et al.* , 1995e; De Gràcia, 1993] provided DAI-DEPUR with a classification of the real *specific working situations* of the concrete plant. It is interesting to notice that with these data LINNEO $^+$ discovered that there exists four subtypes of *normal situations*, not considered before by the experts, but important to describe the behaviour of the process. The experts accepted them and this experiment also revealed that about 20% of the variables provided by the plant's operators were not relevant for the characterization of the *situations* . It is interesting to compare these *situations* with those defined *a priori* in the literature and by the experts, and that are shown in chapter 6 and 8. The new situations we discovered were:

Toxic substances loading Normal (4)

Primary-treatment problems Solid's shock

Plant problems Storm

Secondary-treatment problems

From these new classes, some objects (cases) belonging to each class were selected to be included in the initial Case Library.

5.4.2.2 Case-based Reasoning

The Case-based Reasoning method makes use of the experiential specific knowledge stored in the Case Library (both from observed and experienced cases) to diagnose the current working situation of the WWTP (see figure 5.15), by means of retrieving the most similar case to the current one. Next, the different steps involved in the Case-based reasoning are detailed: case retrieval, case adaptation and evaluation of proposed solutions.

Case retrieval

The task of retrieving cases in the case library is slightly more difficult than typical retrieval in databases. In database systems, the recalling algorithms use an exactly matching method, whereas in a Case Library retrieval, a partial-matching strategy should be used.

The retrieving process of a case (or a set of cases) from the system's memory (the Case Library), usually consists of two substeps:

Searching the most similar cases to the new case: the goal of this stage is recalling the most promising cases based on using some direct or derived features of the new case as indexes into the Case Library.

Selecting the best case(s): the best case(s) among those ones collected in the previous step are selected. Commonly, this selection is made by means of a case ranking process through a similarity or distance function. The best retrieved case is the closest one (most similar) to the new case.

Next, the basic searching algorithm and the distance function used in our system are detailed.

Fig. 5.15. Experiential specific knowledge

a) Basic searching algorithm

Indexes used in the search process are chosen from the prioritized discriminant list of attributes defined by the experts. Although there exists, in the literature, some approaches to automatically obtain these lists of attributes based on the results of a clustering system [Belanche and Cortés, 1991; Baim, 1988], in this case we prefer the expert's opinions. The list must be usefully predictive and discriminant. Our indexing method is based on a predictive discriminating checklist.

The basic searching algorithm can be summarized as a recursive function as follows:

The initial call to search-cases is: search-cases (*new-case* root(*case-library*) 2).

The searching algorithm has three options at each node of the tree:

If the node is a *null node*, it means that the case library is empty, so no case is retrieved.

If the node is a *terminal node* , then the algorithm returns the list of cases of that leaf.

If the node is a *non-terminal node*, then the two best childs of the node are computed. If the new case has a value for the attribute of the node, then the best two childs are the most similar values (one-dimensional matching) for the attribute, but if the new case has a *missing value* for the attribute, then the best childs are the two arcs with higher frequency values.

The two kinds of searching the case library are included in order to overcome a second main problem with discrimination trees/networks: a wrong choice at a high node as an effect of the discretization process of the attribute's values, *etc.*, or if the hierarchy of nodes does not correspond to the importance of the features, as for example a bad discrimination order of the attributes, *etc.* can provoke that some potentially good cases will not be reached in the retrieval process. To cope with these possible troubles, additional partial-matching flexibility is provided in the searching algorithm.

Fig. 5.16. Paths explored by the searching algorithm

So, the exploration-type 2 means that the two best childs of the current node will be explored. The best child is explored again with exploration-type 2, and the second best child is explored with exploration-type 1, if possible. The exploration-type 1 means that only one path (the best one) will be explored.

Thus, the cases retrieved are all the cases stored in the memory that differ at most in one attribute's value from the new case (see figure 5.16). If we call n to the maximum number of discriminant attributes, then at most n paths from the root are explored, and at most n^2 nodes are explored. So, the searching time ($T(n)$) is upper bounded by a function of the number of discriminant attributes (usually small) and does not depend on the number of cases stored in the case library (usually bigger as the system grows): $T(n)\ O(n^2)$.

b) Case selection: a ranking distance function

Selecting the best similar case(s) or previous working situations, it is usually performed in most Case-based reasoners by means of some evaluation heuristic functions or distances, possibly domain dependent. The evaluation function usually combines all the partial-matchings through a dimension or attribute of the cases, into an aggregate or full-dimensional partial-matching between the searched cases and the new case. Commonly, each attribute or dimension of a case has a determined importance value (weight), that is incorporated in the evaluation function. This weight could be static or dynamic depending on the Case-based reasoner purposes. Also, the evaluation function computes an absolute match score (a numeric value), although a relative match score between the set of retrieved cases and the new case can also be computed.

Most Case-based reasoners such as REMIND [Cognitive, 1992], MEDIATOR [Kolodner and Simpson, 1989], PERSUADER [Sycara, 1987], *etc*. use a generalized weighted distance function such as,

$$\text{Full-dist}(C_i, C_j) = \sum_{k=1}^{n} w_k * \text{atr-dist}(C_{ik}, C_{jk}) \Big/ \sum_{k=1}^{n} w_k$$

but this kind of distance functions, sometimes, does not capture the significant differences among the attributes, because they are a *lineal* combination of one-dimensional distances. We have assumed that a *non-lineal* multi-dimensional distance function would be required for a better matching performance. After a wide performance study, we have developed a *normalized exponential weight-sensitive distance function* , that we have called the *Eixample distance*. It takes into account the different nature of the quantitative or qualitative values of the lineal (ordered) attributes, and the modalities of categorical (not ordered) attributes.

Eixample distance is sensitive to weights, in the sense that, for the most important attributes, that is weight > a, the distance is computed based on their qualitative values, i.e. maintaining or amplifying the differences between cases, and for those less relevant ones, that is weight a , the distance is computed based on their quantitative values, i.e. reducing the differences between cases.

The *Eixample distance* used to rank the best cases is:

n n

$$d(C_i, C_j) = e^{Wk} * d(A_{ki}, A_{kj}) / e^{Wk}$$

k=1 k=1

where,

$d(A_{ki}, A_{kj}) = |quantval(A_{ki}) - quantval(A_{kj})| / (upperval(A_k) - lowerval(A_k))$

if A_k is a lineal (ordered) attribute and W_k a

$d(A_{ki}, A_{kj}) = |qualval(A_{ki}) - qualval(A_{kj})| / (\#mod(A_k))$

if A_k is a lineal (ordered) attribute and $W_k > a$

$d(A_{ki}, A_{kj}) = 1 - d_{qualval(A_{ki}),qualval(A_{kj})}$

if A_k is a categorical (not ordered) attribute

and,

C_i is the case i; C_j is the case j; W_k is the weight of attribute k; A_{ki} is the attribute k in the case i; A_{kj} is the attribute k in the case j; quantval(A_{ki}) is the quantitative value of $A_{ki;}$ quantval(A_{kj}) is the quantitative value of A_{kj}; A_k is the attribute k; upperval(A_k) is the upper quantitative value of A_k; lowerval(A_k) is the lower quantitative value of A_k; a is a cut point on the weight of the attributes; qualval(A_{ki}) is the qualitative value of A_{ki}; qualval(A_{kj}) is the qualitative value of A_{kj}; #mod(A_k) is the number of modalities (categories) of A_k.; $d_{qualval(A_{ki}),qualval(A_{kj})}$ is the d of Kronecker.

The system computes the distances between each case retrieved by the discrimination tree search and the new case. Then, it ranks the list of cases by increasing distance. So, the first case in the list is the closest case to the new case. The output of this process is a comparative table of all the retrieved cases and the new case, describing the distance and the attribute's values. An example of this output is shown in table 5.2.

CASE	DIST	O-SS	O-COD	I-SS	I-COD	I-BOD	Q	D-SS	D-COD	I-pH	O-pH
NEW		60	240	480	380	31000	210	320	7.8	7.6	
21	0.000	60	240	480	380	31000	210	320	7.8	7.6	

1	0.018	84	166	407	NIL	44101	94	280	7.8	7.3
8	0.022	NIL	964	483	156	34094	92	170	7.8	7.6
9	0.026	79	1000	457	189	39421	140	323	7.9	7.8
7	0.073	70	172	370	365	32217	135	7.7	7.6	
15	0.199	82	218	517	230	38105	102	349	8.4	8.0

Table 5.2. The table of retrieved cases in the Case-based system

Case adaptation

When the best partial-matching case selected from the case library does not match perfectly with the new case, the old solution needs to be adapted to fit more accurately the new case solution. This *reusing* process can happen during the solution formulation (*adaptation*), or after some feedback has pointed out some problem in the evaluation step, that needs to be fixed(*repair*).

There are a lot of strategies that have been used in the Case-based reasoners. All these techniques can be grouped [Kolodner, 1993; Riesbeck and Schank, 1989] as *null adaptation* , *structural adaptation* and *derivational adaptation* , although in most Case-based reasoners, several mixture kinds of adaptation methods are implemented.

Null adaptation could be a right strategy in Case-based systems with very simple actions in the solution (like accept/reject, a fault diagnosis, *etc.*) such as the first adaptation method used in the PLEXUS system [Alterman, 1988]. In that systems, the old solution is applied directly to the new case.

There are several *structural adaptation* methods, where the adaptation process is applied directly to the solution stored in a case. The structural adaptation methods can be divided in three major techniques: *substitution methods* , *transformation methods* and *special-purpose adaptation heuristics* or *critic-based adaptation methods* .

Substitution methods provide the solution of the new case with appropriate components or values computed from components or values in the retrieved solution. Most outstanding substitution techniques are: *parameter adjustment* or parameterized solutions, where the differences between the values of the retrieved case and those ones of the new case are used to guide the modification of the solution parameters in the appropriate direction. This approach has been used, for example, in HYPO [Ashley, 1990] and PERSUADER [Sycara, 1987], JUDGE [Bain, 1986]; another kind of methods, such as *direct reinstantiation* used in CHEF [Hammond, 1989], *local search* used in JULIANA [Shinn, 1988], PLEXUS and SWALE [Kass and Leake, 1988], *query memory* used in CYRUS [Kolodner, 1985] and JULIANA, *specialized search* used in SWALE, *etc* ., can be named as *abstraction and respecialization* methods. When there is a component (object, value, *etc* .) of the retrieved solution that do not fit the new problem, these methods look for abstractions of that component of the solution in a certain knowledge structure (concept generalization tree, *etc*.) that do not have the same difficulty; the last kind of substitution methods is the *case-based substitution* methods. They use the differences between the new and the retrieved case to search again cases from the case library to eliminate these differences. These techniques have been used, for instance, in systems such as CLAVIER [Hennessy and Hinkle, 1992], JULIA [Hinrichs, 1992], CELIA [Redmond, 1992], *etc*.

The *transformation methods* uses either some *common sense transformation* rules, like deleting a component, adding a component, adjusting values of a component, *etc*.), such as in JULIA system, or some *model-guided repair transformation* techniques based on a causal knowledge, such as in KRITIK [Goel and Chandrasekaran, 1992] or CASEY [Koton, 1989] systems.

The *special-purpose adaptation techniques* or *critic-based adaptation methods* are based on some specific rules of repairing, called *critics* [Sacerdoti, 1977; Sussman, 1975], like those used in PERSUADER. Other systems such as CHEF and JULIA, use some domain specific adaptation heuristics and some structure modification heuristics.

Derivational adaptation methods do not operate on the original solutions, but on the method that was used to derive that solution. The goal is rerunning the same method applied to derive the old solution, to recompute the solution for the new case. This methodology was first implemented in ARIES system, and was named as *derivational replay* [Carbonell, 1986]. In such a techniques, reinstantiation occurs when replacing a step in the derivation of the new solution, like in systems such as PRODIGY/ANALOGY [Veloso and Carbonell, 1993], JULIA [Hinrichs, 1992] or MEDIATOR [Kolodner and Simpson, 1989].

In WWTP domain, the solutions stored in the cases are composed by very simple operation actions, such as increase the dissolved oxygen (DO) set-point, decrease the DO set-point, increase the recirculation flow, *etc*. (see chapter 7). The main task in the supervision and control problem is searching a similar diagnosis (operating situation) for the current situation in a concrete plant. If the retrieved case is close enough to the new case, the old solution (*i.e.* , the control actions) only need a few adaptation changes. Thus, we can assume that most times the retrieved cases are so similar to the new ones, that they only need *parameter adjustment* to derive the new solution. For each important difference between the attributes of the new case and the retrieved one, the system adjusts the parameters pointed by these differences. The adjustment in this stage of DAI-DEPUR is done by interpolation. In other cases, special-purpose adaptation heuristics must be developed. Our approach to the adaptation process can be summarized as follows:

if distance(C_{NEW}, C_{RETR}) b **then**

solution-parameters-interpolation(C_{NEW}, C_{RETR})

else

special-purpose-adaptation-heuristics(C_{NEW},C_{RETR})

endif

where b is a cut point on the distance value, b [0.2, 0.3]

Evaluation of proposed solutions

This step is one of the most important for a Case-based reasoner. It gives the system a way to evaluate its decisions in the real world allowing it to receive feedback that enables it to learn from success or failure.

Evaluation can be defined as the process of assessing the goodness or performance of the proposed solution for the new case, derived from the solution of the best similar remembered case. The evaluation process can point out the need for additional adaptation – usually called *repair*– of the proposed solution, although this only make sense, in non real-time world domains. Commonly, this evaluation step can be performed either by *asking* to a human expert (oracle) whether the solution is a good one or not, or by *simulating* the effects of the proposed solution on the real world such as in most planning or design domains, or by directly *getting a feedback* on the results of the proposed solution, from the real world.

In our system, the evaluation is given by a human expert, i.e. the plant's manager or operator, as we are in a development phase of the system. Notwithstanding, in middle-term future the evaluation will be performed by testing the effectiveness of the given solutions to the wastewater treatment plant operation. First, we plan to assess our system with a pilot scale plant already built-up and operating since two years ago. Afterwards, we plan to check the Case-based system on real plants.

5.4.2.3 Learning from Experience

Learning is an interesting and essential cognitive task of the Case-based systems. Mainly, there are two major kinds of learning in a Case-based system: learning by observation and learning by own experience. *Learning by observation* happens when the system is provided with a set of initial cases, either by an expert or by direct observation (experientiation) of real data, as it is described in 5.4.2.1. Also, it can learn a new case by direct observation provided by an expert in any moment.

Learning by own experience is being done after each cycle of the Case-based reasoner. After an evaluation step appears the opportunity to increase the problem solving capabilities of the system. So, it can learn from the new experience. If the proposed solution has been a successful one, the system can learn from this fact, in the sense that if this experience is stored in memory, when a new similar case to this one appears, it can be solved as the past one (*learning from success*). If the system has failed, it must be able to prevent itself from making the same mistake in the future (*learning from failure*). Not all the Case-based systems have both kinds of learning.

Learning from success

When the new case has been successfully solved, the main option to follow is to store this new case in the case library. So, this task means to insert the new experience in the appropriate place into the case library, so that it can be remembered when it can be more useful, and cannot be recalled indiscriminately. In other words, the case must be placed in the neighbourhood space of the memory where it would be easily recalled in the retrieval step. Thus, good indexes must be chosen to implement this strategy. While the new case is being placed in the case library, memory's indexing structure and organization is updated appropriately.

The storing task of new cases can be summarized as a recursive procedure as described in the next algorithm, with an initial call to the algorithm:

learn-case (new-case root(*case-library*) discr-list-of-attrib)

The storing algorithm has three main options at each node of the tree:

If the node is a *null node* , i.e., the case library is empty, then the algorithm builds up a new leaf-node with the case.

If the node is a *terminal node* , then it computes if there exists a next discriminant attribute for the leaf-case and the new case. If there is not such a new attribute, then it adds *selectively* the case to the node. Our selective criterion is that the number of cases stored on that node does not exceed an upper bound. We are developing a new strategy based on a *relevance* *measure* of the new cases to be added. If there exists a new discriminant attribute, there are four options depending on whether the leaf-node and the case have a null value (nil) or not. In three options, the algorithm recursively continues the storing task on a new-node (with the case) or the leaf-node. The other option makes two childs from the node with the case and the leaf-case.

If the node is a *non-terminal node*, the algorithm checks the branch with the same value as the current case. If this branch exists, it calls recursively to the procedure with that child as a node. If this branch does not exist, and the value of the current case is not null (nil) then it opens a new branch with that value. If the value is nil, then it recursively continues the procedure with the most frequently used child-node.

Another issue that is taken into account in our system, in order to provide it with a criterion for a *case forgetting* policy (as it is described in 5.4.2.4), is the utility of cases. We have used (as other Case-based systems) a predictive measure of the utility of cases in solving future experiences. Each time a relevant case has been successful, the utility measure of all retrieved cases is updated. The *normalized utility measure* we are using is computed as:

$$UM(C) = ((\#\,UaS / \#\,S) - (\#\,UaF / \#\,F) + 1) / 2$$

where,

C is a retrieved case; # UaS is the number of times that the case was Used and there was a Success, when the case was retrieved; # S is the total amount of Successes when the case was retrieved; # UaF is the number of times that the case was Used and there was a Failure, when the case was retrieved; # F is the total amount of Failures when the case was retrieved. Thus, when the UM(C) is 0 means that the case C is very unuseful, and on the other hand, when it is 1 means that the case C is very useful.

This utility measure combines the usual measure of the frequency value of the successes of the case with a *reinforcement measure* of the frequency value of the failures of the case.

Learning from failure

Assuming that the adaptation method is correct, two reasons could originate a failure. One is that the case retrieved is the best one for solving this new situation, given the current case library, although it is not very similar. The problem here is that there are not enough cases (experience) in the case library to cover the whole space of cases. The solution relies on learning a lot of new relevant experiences to store in the case library. The other reason is that although it exists a very similar case to the new one, it has not been retrieved. Thus, there is something wrong in the retrieval process. Perhaps the distance function is not correct because the strategic importance of attributes (weight) is not well suited or the function do not capture accurately the differences between qualitative and quantitative values of attributes. Perhaps what is wrong is the discrimination tree organization. If the discriminating list of attributes provided by the experts is not good, then the retrieval algorithm can miss the best similar case due to the fact that it will be searching in some other region of the tree, where the best similar case is not there. The solution to this problem relies in re-organizing the case library's structure.

When the new case has failed, there are several possible actions to be taken, in order to ensure that this failure cannot be repeated in future. First, the Case-based system can store the failed case into its memory to prevent taking, another time, the same failed solution for similar cases to this one. Some Case-based systems maintain a separate case library of failed cases, and others maintain only one case library structure.

In the first systems, a previous step is added to the general Case-based reasoning cycle: the *anticipation phase*. Before retrieving any successful case in memory, it is recalled whatever case in the failed case library, that matches the input new case to avoid repeating the failure. In the other systems, the equivalent of that anticipation step is implemented as a *filtering task* applied to the searched cases in the retrieving process. They eliminate the previous failed cases and the cases that were the source experience to derive those failed cases, from the list of retrieved cases. Thus, the system can avoid to make the same incorrect action that in past situations.

Another interesting action to do when there is an available human expert, is incorporate to the system's memory the right solution that he proposes to solve the new situation. So, in the future, this recorded experience could be remembered and used appropriately.

Other tasks that can be performed are the updating of the weight of the attributes, then modifying the distance function, or changing their order in the discriminating list. Another feature is to update the utility measure of the retrieved cases that could derive the new case.

In our Case-based system, the failed cases are stored together with the good ones, in the Case Library. A mark about its failure needs to be signalled, in order to be used as a filtering feature in the searching process. In the next future, DAI-DEPUR will be running on a real WWTP, where it is not sure that a human expert will be available all the time (*i.e.* , Cassà de la Selva-Llagostera WWTP). Thus, the system as a default strategy cannot incorporate the correct case provided by an expert. Also, the system recomputes the utility measure of the retrieved cases.

5.4.2.4 Introspection

As Case-based system learns and evolves over time such as the human beings, it can reason about itself (*reflection* or *introspection*), about its cognitive tasks and about its memory organization to refine all these features, as stated in [Fox and Leake, 1995].

So, it is possible that its *performance* (efficiency in time) get worse due to the fact that the time spent in the retrieval step is becoming greater. Usually, the organization of the case library is the origin of this problem. Sometimes, in hierarchical structures as in the discrimination trees, appear some straight paths that ends with a leaf-node. In these paths, there are no branches at all. If the tree can be arranged in such a way that the depth of the tree become smaller than the current one (*more compact*), then the average retrieval time of cases can be reduced.

Also, the size (efficiency in space) of the case library can increase very much with the storing of redundant cases, without an extensive improvement of the performance of the system, as pointed in [Miyashita and Sycara, 1995]. A natural human cognitive task appears as the solution to this trouble: *forgetting*. Human beings forget the knowledge that they do not use. And hopefully, what it is not used is what is not useful for their goals. Bearing in mind this analogy, we claim that there are not very useful cases stored in the memory of the Case-based systems that can be removed, with a significant increase of the performance. The *unuseful* cases are those with a low utility measure (UM(C) < d) and that are (partially) *redundant* and that are not exceptional ones. *Exceptional* cases are that cases significatively different (distantiated) from the most of the stored cases. Although they can be used few times, it is necessary that they stay at the memory for the *competence* issue of the system.

We are implementing these techniques in our Case-based system within a reflective cycle, to guide the case library towards an optimal configuration of cases, maximizing *competence* and minimizing size and response-time (*performance*) of the system.

Chapter 6

The Situations Level

6.1 Introduction

A crucial point for the WWTP supervision is the concept of working situations. As it has been defined previously in chapter 3, *a **situation** is an operational working state of the plant, described by measures of the relevant attributes of the process*. In the next sections of this chapter we will detail the domain models used at this level: the generic situations defined by the experts (*expert generic situations*, from now on named generic situations) and the specific situations occurred in a concrete WWTP (*experiential specific situations*, from now on named specific situations). The main task involved within this level is the supervision task, implemented through the combination method, to identify the current global situation of the WWTP.

6.2 Domain Models

Two complementary domain models are used in DAI-DEPUR at this level, to capture the working situation of a WWTP: the *specific situations* and the *generic situations*.

6.2.1 Generic Situations

In order to obtain the main *generic situations* considered by the experts, they were required to define the usual situations that they thought as relevant for WWTP's operation. The situations were defined by means of some values of the data taken into account by DAI-DEPUR.

The generic situations that DAI-DEPUR takes into account both coming from the experts and from the wastewater treatment plants literature, can be grouped in four major types:

- *Operational problem situations* which group any bad operating situation that can be solved with a correct actuation over the WWTP. The main situations are Foaming, Rising, Bulking, Non mechanical primary settlers problems, Bioreactor anaerobic conditions, Sidestreams.

- *Mechanical or accidental problem situations* which include any abnormal situation caused by a mechanical or accidental fault such as Mechanical primary settlers problems, Low primary settlers wasting, Bad primary settlers wasting temporization, Bad closed by-pass, Anomalous pH or DO sensor signalling, Turbines crash, Broken clarifier bridge rotatory band, Electrical power off.

- *Future problem transition situations* which group any situation that in a middle-term will probably lead the WWTP to an abnormal state, i.e. a transition situation. Abnormal pH, Abnormal environmental temperature, Anomalous sludge retention time (SRT), Anomalous F/M ratio, High secondary settlers sludge level.

- *Abnormal inflow situations* are any situation in which some of the inflow characteristics are different from usual ones. For example a Grease inflow, Inflow sulphurs, Toxic shock, Loading shock, Solids shock, Storm, Overflow.

In next paragraphs, the main generic expert situations considered are briefly explained:

Rising

Rising sludge is a flotation process caused by bubbles of free nitrogen adhering to the sludge flocs. This is typically observed in secondary settlers of activated sludge plants with nitrification or partial denitrification. Clumps of sludge already settled will rise to the surface and break up into smaller parts, some of which will partially settle again while others will escape from the settler with the overflow. Nitrogen bubbles can be seen in the water at the top of the settler. The result is increased suspended solids in the effluent and an overall reduction of effluent quality with respect to BOD, COD and total nitrogen and phosphorus.

The only practical solution to the problem of rising sludge in treatment plants with summer temperatures above 20 ºC seems to be denitrify the wastewater before it enters the settler in order to ensure a nitrate influent concentration below the critical one [Henze *et al.*, 1993].

Foaming

Two main types of scum may appear in activated sludge plants:

A white-grey foam occurs on aeration basins and in secondary effluents when a plant is being operated at just about the upper limit of F/M (start-up of the plant).

A viscous, stable and often chocolate-coloured foam has been associated with the presence in the activated sludge of large number of filamentous bacteria which possess poorly wettable cell surfaces. They render the flocs hydrophobic and amenable to attachment on air bubbles. The air bubble-floc aggregate is less dense than water and therefore floats to the surface.

Bulking

Bulking is due to filamentous organisms that provide the macrostructure of the floc are present in large numbers. These filamentous organisms extend from flocs into the bulk solution and interfere with compactation, settling, thickening and concentration of the activated sludge. The type of interference depends on the causative filamentous organism. There are several kinds of possible Bulking causes:

- Low dissolved oxygen (DO) Bulking
- Low Sludge Retention Time (SRT) Bulking
- Nutrient deficiency Bulking
- Sulphur Bulking

- F/M Bulking

Non-mechanical problems in primary settlers

The nature of Non-mechanical problems in primary settlers and its severity are frequently temperature dependent. In cold weather, sludge will be more difficult to pump, sludge lines will collect grease more quickly, and skimming quantities will increase; however, septicity and odour will likely diminish.

Bioreactor anaerobic conditions

Whatever kind of problem related to the DO level in the bioreactor such as an inadequate aeration motor switches schedule, a turbine mechanical problem, or an incorrect DO sensor behaviour can lead some portion of the mixed liquor to remain in anaerobic conditions.

Sidestreams

Most of the processes in the plant produce a little quantity of dirty water called sidestreams. Normally, these sidestreams are not treated separately but are returned to the plant headworks. The high strength and intermittent flow of these sidestreams can adversely affect the performance of some small plants because, although generally low in suspended solids, this sidestream flow can contain large quantities of soluble BOD and nutrients, particularly from sludge processing.

Abnormal pH

As in whatever biological process, the microorganisms present in the bioreactor have an activity optimal range of pH. Abnormal pH values directly affect the substrate removal efficiency in the bioreactor.

Abnormal Environmental Temperature

As in whatever biological process, the microorganisms present in the bioreactor have an activity optimal range of temperature. Colder temperatures directly affect the substrate removal efficiency in the bioreactor.

Anomalous sludge retention time (SRT)

The sludge retention time determines the microbial population present in the bioreactor. Thus, an anomalous sludge retention time implies the growth of undesirable microorganisms which avoid the correct achieving of the goals of the biological process.

Anomalous F/M ratio

For a correct operation of WWTP it is necessary to keep a balanced F/M ratio. High or low ratio values causes the appearance of filamentous microorganisms which will origin the well known separation problems in the secondary settlers, i.e. bulking.

High secondary settlers sludge level

High stored quantity of sludge in the secondary settlers can derive in a loss of biomass with the effluent.

Grease inflow

The presence of grease in the inflow favours the appearance of filamentous microorganisms such as *Nocardia*, which can derive in a serious foaming problem.

Inflow sulphurs

The presence of grease in the inflow favours the appearance of filamentous microorganisms such as *Beggiatoa*, which can derive in a serious bulking problem.

Toxic shock

The presence of toxicants in the effluent wastewater, especially heavy metals, can result in dispersed growth due to deflocculation. Filamentous organisms are often the first microorganisms to be affected by toxic metals. The SVI decreases rather rapidly. If the toxicity is severe enough or reoccurring, defloculation will occur. If the toxicity even is severe enough, protozoa are killed and their lysed cell contents can cause a foam. The activated sludge BOD removal usually declines or ceases following this event.

Loading shock

This situation is characterized by a high level of organic matter in the inflow that can quickly vary the F/M ratio and WWTP operation must be changed to decrease the ratio overcoming this shock.

Solids shock

This situation is characterized by a high level of suspended solids in the inflow that can plug the pretreatment grids and/or overload the primary settlers.

Storm

When heavy flows originated by storms reach the activated sludge they can sweep a large amount of solids along with them, thereby producing low quality effluent and putting the biological process out of order.

	Deficient primary sludge exit	Deficient primary sludge settling	Non alternated sludge purge	Heavy metal accumul.	Heavy metal peak	Bio-reactor anaerobic condition
Rotatory band	-	-	-	-	-	-
Bubbles secondary settler	-	-	-	-	-	-
Bubbles primary settler	yes	-	yes	-	-	-
Inflow	-	-	-	-	normal	-
Dissolved Oxygen (DO)	-	-	-	high	normal	low
I-COD	-	-	-	-	normal	-
O-COD	-	-	-	high	high	high
SRT	-	-	-	-	normal	-
Bio-foam	-	-	-	-	-	black
Filament.	-	-	-	-	-	-
SVI	-	-	-	-	normal	-
Odour	-	-	-	-	-	yes
Sulphurs	-	-	-	-	normal	-
Primary sludge exit	deficient	normal	normal	-	normal	-
O-SS	high	high	high	-	normal	-
B-SS	high	high	high	-	normal	-
I-SS	-	-	-	-	normal	-
B-VSS	-	-	-	-	normal	-
R-VSS	-	-	-	-	normal	-
Temp.	-	-	-	-	-	-
Turbines	-	-	-	-	following schedule	following schedule
B-heavy metal	-	-	-	high	-	-
I-heavy metal	-	-	-	-	high	-

Table 6.1 (a). Definition of the generic expert situations

Overflow

It is a situation strongly related to the storm situation. It is caused by a very strong storm in a short period of time. Its consequences are similar but more serious than the storm situation.

	Turbines crash	Foaming	Bad closed by-pass	Heavy metal Bulking	Low DO Bulking	Sulphurs Bulking
Rotatory band	-	-	-	-	-	-
Bubbles secondary settler	-	-	-	-	-	-
Bubbles primary settler	-	-	-	-	-	-
Inflow	-	-	-	-	-	-
Dissolved Oxygen (DO)	low	slightly high	-	-	low	-
I-COD	-	-	-	-	-	-
O-COD	high	high	high	high	high	high
SRT	-	-	young	-	-	-
Bio-foam	-	brown	-	-	-	-
Filament.	-	high	-	normal	high	high
SVI	-	bulking	-	bulking	bulking	bulking
Odour	yes	-	-	-	-	-
Sulphurs	-	-	-	-	-	high
Primary sludge exit	-	-	-	-	-	-
O-SS	-	-	-	-	-	-
B-SS	-	-	low	-	-	-
I-SS	-	-	-	-	-	-
B-VSS	-	-	low	-	-	-
R-VSS	-	low	-	low	low	low
Temp.	-	-	-	-	-	-
Turbines	broken	-	-	-	-	-
B-heavy metal	-	-	-	high	-	-
I-heavy metal	-	-	-	-	-	-

Table 6.1 (b). Definition of the generic expert situations

	F/M Bulking	High DO Bulking	Non Filament. Bulking	Rising	Loading shock	Side-streams loading shock
Rotatory band	-	-	-	-	-	-
Bubbles secondary settler	-	-	-	yes	-	-
Bubbles primary settler	-	-	-	-	-	-
Inflow	-	-	-	-	-	-
Dissolved Oxygen (DO)	-	high	-	-	low	low
I-COD	low	-	-	-	high	low
O-COD	high	high	high	high	high	high
SRT	-	-	old	old	-	-
Bio-foam	-	-	-	-	-	-
Filament.	-	normal	normal	normal	-	-
SVI	bulking	bulking	bulking	normal	-	-
Odour	-	-	-	-	-	-
Sulphurs	-	-	-	-	-	-
Primary sludge exit	-	-	-	-	-	-
O-SS	-	-	-	high	-	-
B-SS	-	-	-	high	high	high
I-SS	-	-	-	high	-	-
B-VSS	normal	-	-	normal	high	high
R-VSS	low	low	low	-	-	-
Temp.	-	-	-	-	-	-
Turbines	-	-	-	-	-	-
B-heavy metal	-	-	-	-	-	-
I-heavy metal	-	-	-	-	-	-

Table 6.1 (c). Definition of the generic situations

In table 6.1 (a), (b) and (c) there are described the definition of some of these general situations based on the local Manresa's WWTP, by means of some data values.

That situations can be clustered by means of a classification method, such as LINNEO$^+$ (our knowledge acquisition method), providing DAI-DEPUR with an *associative model* of the global situations of a WWTP. See table 6.2 for a possible classification obtained.

CLASS	SITUATIONS
c1	Heavy metal accumulation
c2	Bad aeration motor switches schedule
c3	Deficient primary sludge exit Deficient primary sludge settling Non alternated primary settlers purge
c4	Heavy metal Bulking High dissolved oxygen Bulking Non Filamentous Bulking
c5	Sulphurs Bulking Low dissolved oxygen Bulking F/M Bulking
c6	Turbines Crash Bioreactor anaerobic conditions
c7	Foaming
c8	Overloading
c9	Normal Heavy metal peak
c10	Rising Settler rotatory band broken
c11	Storm
c12	Loading shock Side-streams loading shock
c13	Low loading
c14	Bad closed by-pass

Table 6.2 Classification results

6.2.2 Specific Situations

The *specific situations* are those situations actually experienced in the WWTP under supervision. These specific situations are stored as cases in a Case Library memory, to serve as the basis for the Case-based Reasoning method for a complementary diagnosis task.

As it has been explained in 5.4.2.1, the initial Case Library was fed by some real cases extracted from a classification (see appendix C) of real operation data of Manresa's WWTP, provided by LINNEO$^+$. These initial experiential situations are:

CASE	DAY	Q	I-Zn	I-pH	I-BOD	I-COD	I-SS	D-COD	D-SS	O-pH	O-COD	O-SS
C-1	1/3/90	44101	1.5	7.8	NIL	407	166	280	94	7.3	84	21
C-2	29/6/90	35198	0.8	7.7	185	372	164	278	74	7.9	78	15
C-3	13/3/90	42393	0.7	7.9	189	478	230	412	104	7.6	306	131
C-4	14/3/90	42857	1.5	7.7	238	319	292	276	104	7.4	350	238
C-5	18/7/91	NIL	1.0	7.6	186	495	222	NIL	112	7.6	292	74
C-6	1/10/90	47623	3.4	7.7	183	310	170	227	108	7.9	85	20
C-7	31/3/91	32217	2.0	7.7	165	370	172	135	80	7.6	70	16
C-8	31/5/91	34094	0.9	7.8	156	483	964	170	92	7.6	NIL	20
C-9	5/6/90	39421	1.0	7.9	189	457	1004	323	140	7.8	79	21
C-10	29/4/90	27333	2.0	7.6	238	348	174	364	104	7.4	210	73
C-11	14/9/90	41206	3.3	7.8	117	366	500	181	106	7.8	67	12
C-12	1/4/91	34573	0.7	7.7	156	276	146	176	124	7.7	43	44
C-13	5/8/91	29719	0.2	7.6	133	284	186	160	52	7.8	60	21
C-14	12/8/90	47718	0.7	7.8	31	81	208	80	233	7.9	25	12
C-15	1/2/91	38105	1.5	8.4	230	517	218	349	102	8.0	82	13
C-16	23/11/90	28819	2.2	8.4	195	392	188	267	82	7.9	86	19
C-17	22/10/90	48950	2.5	8.1	109	211	880	111	118	7.9	35	13
C-18	24/5/91	36495	0.1	7.7	213	627	2008	226	66	7.6	119	13
C-19	15/3/90	42911	0.7	7.6	114	252	116	216	70	7.5	172	104

Table 6.3. Initial specific situations

6.3 Supervision Task

The supervision task is mainly achieved by the Supervisory-KBS agent, with the collaboration of the WWTP operator. This supervision implies than all the local situations diagnosed by the different distributed KBS at the knowledge/expertise level, have to be combined to get a global insight of the *current generic situation* of the WWTP diagnosed by the Rule-based Reasoning method. Also, the most similar situation stored in the Case Library is retrieved to provide a complementary diagnosis from an *specific situation* point of view.

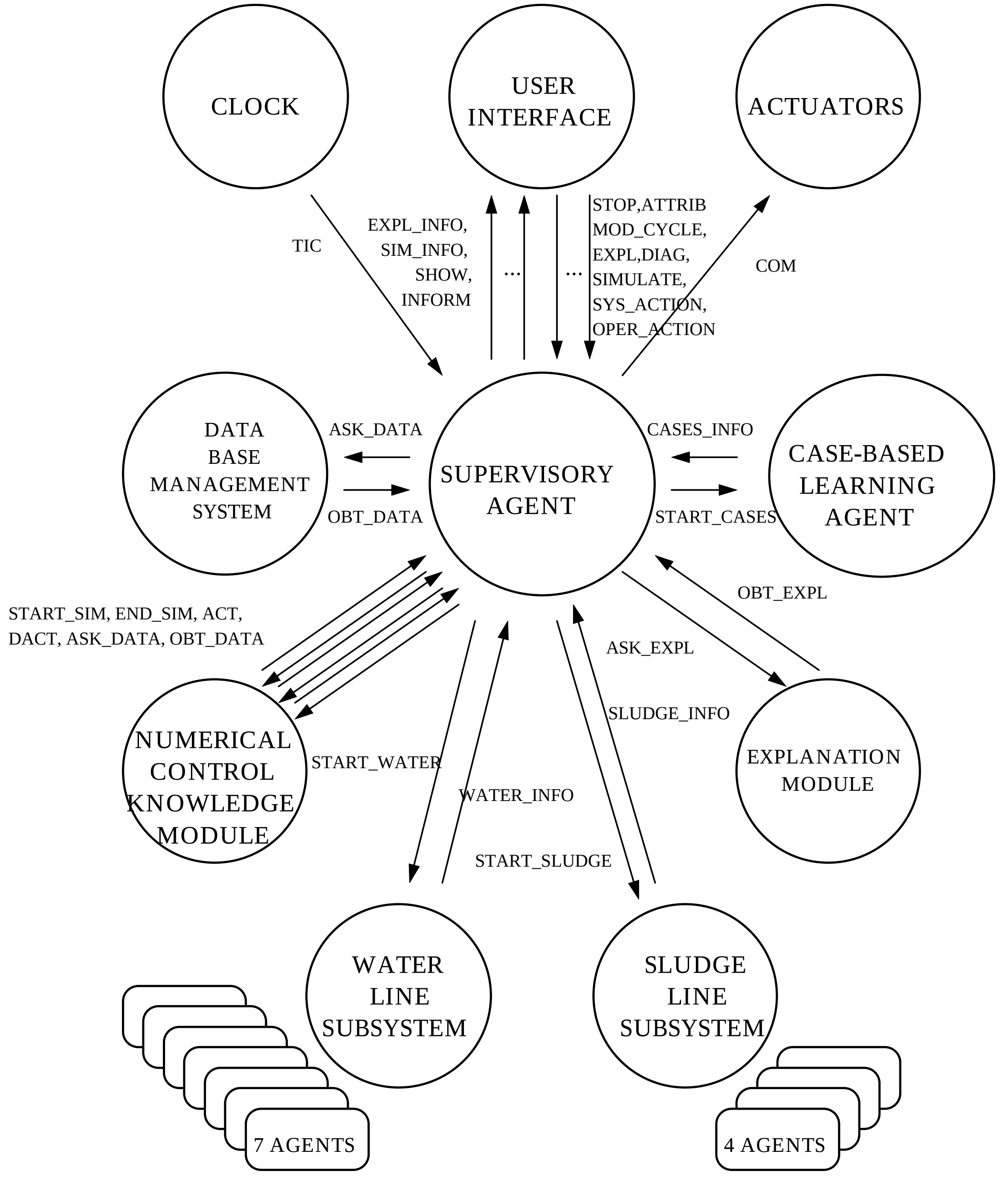
The *Supervisory–KBS* agent is the manager of the distributed problem solving architecture and acts as a master. It receives diagnosis information from the Water line subsystem, from the Sludge line subsystem, and from the most similar case retrieved by Case-based Learning and Reasoning agent. The Supervisory agent notifies both the specific and generic situations to the operator of the plant, who must identify the current working situation. Also it will supervise at the next level (actuations level) the actuation of DAI-DEPUR.

The Supervisory-KBS agent behaviour can be logically summarized as a control loop algorithm using Dijkstra's guarded commands [Dijkstra, 1975] and Communicating Sequential Processes (CSP) [Hoare, 1978] structures:

```
SUPERVISORY AGENT ::
[ var end, water-line_analysis, sludge-line_analysis, case_exploring,
         w, s, c : boolean;
      bound, new_bound, cycle : integer;
      water-line_situation, sludge-line_situation, global_situation : situation;
      cases : list(case); commands : command
  endvar

    bound:= …;      {minutes}
    end:=false;
    cycle:=bound;
    water-line_analysis:=false;
    sludge-line_analysis:=false;
    case_exploring:=false;
    w:=false;
    s:=false;
    c:=false;
    *[¬ end –>
                                          {Monitoring}

      [   tic ?  –> cycle:=cycle+1
      []  cycle _ bound –>  water-line_analysis:=true;
                            sludge-line_analysis:=true;
                            case_exploring:=true;
                            cycle:=0
```

```
                 {Diagnose or evaluation phase}

[] water-line_analysis; start_water ! –> water-line_analysis:=false
[] sludge-line_analysis; start_sludge ! –> sludge-line_analysis:=false
   ...
[] water_info ? water-line_situation –> save–sit(water-line_situation);
                                         w:=true
[] sludge_info ? sludge-line_situation –> save–sit(sludge-line_situation);
                                         s:=true

                 {Learning and Reasoning phase}

[] case_exploring; start_cases ! –> case_exploring:=false
[] cases_info ? cases –> save(cases); c:=true

              {Supervisory and communication phase}

[] w ^ s ^ c –> global_situation:=combine(sludge-line_situation,
                                           water-line_situation);
                solutions:=infer(global_situation, cases);
                w:=false; s:=false; c:=false;
                inform–operator(global_situation);
                inform–operator(solutions)

            {Actuation phase and user interface support}

[] operator_action ? commands –> com ! commands;
                                   save–case(global_situation,commands)
[] System_action ? –> com ! solutions;
                      save–case(global_situation,solutions)

[] stop ? –> end:=true
[] mod_cycle ? new_bound –> bound:=new_bound; cycle:=0
[] attrib ? set_of_attrib –> show(set_of_attrib)
[] expl ? affair –> dem_expl ! affair
[] diag ? –> show–diagram
[] simulate? time –> start_sim ! (time,global_situation)
```

```
    ...
[]  obt_expl ? explanations –> ans_expl ! explanations
[]  end_sim ! results –> ans_sim ! results ]]]
```



START_WATER: Start water line

WATER_INFO: Receive water line diagnosis

ASK_DATA: Ask for some

OBT_DATA: Obtain the data

ACT: Activate the DO control

DACT: Deactivate the DO control

START_SIM: Start

END_SIM: End

. . .

. . .

Fig. 6.1. Distributed problem solving interaction

The distributed problem solving interaction is depicted in figure 6.1.

6.4 Combination Method

The combination method is used twice at this level. First, the different local diagnostics provided by the distributed KBS about the status of the several operational units that compose a WWTP (primary settlers, secondary settlers, bioreactor, recirculation, grit removal, thickening, *etc.*), must be integrated to get the global *generic situation* of the plant. This is achieved by the Rule-based Reasoning method, from the knowledge embodied in the Supervisory-KBS. See figure 6.2 for an example of a set of combination rules (global diagnosis rules)

SUPERVISORY-KBS-RULE-002 ::
IF I-COD-status is normal AND O-COD-status is high
THEN conclude that
COD Removal efficiency is low

SUPERVISORY-KBS-RULE-037 ::
IF COD Removal efficiency is low AND B-VSS-status is high
THEN conclude that
SIDESTREAMS situation is VERY-POSSIBLE

SUPERVISORY-KBS-RULE-017 ::
IF Bioreactor-pH-status is normal AND presence-fungi is true
THEN conclude that
Bioreactor-pH-sensor is bad ALMOST-SURE

SUPERVISORY-KBS-RULE-036 ::
IF B-VSS-status is high AND Wasting-flow-status is normal AND
COD Removal efficiency is normal
THEN conclude that
wasting-pipe is plugged VERY-POSSIBLE

SUPERVISORY-KBS-RULE-042 ::
IF Input-pH-variability is true
THEN conclude that
Fear-of bulking is true

SUPERVISORY-KBS-RULE-049 ::
IF Inflow-status is high AND Inflow-status is high during the last hour
THEN conclude that
STORM situation is VERY-POSSIBLE
AND
THEN conclude that
OVERFLOW situation is POSSIBLE

Fig. 6.2 An example of combination rules

On the other hand, the most similar experience occurred in the concrete WWTP, supplies DAI-DEPUR with a *specific situation* diagnosis, obtained by the Case-based Reasoning method. See table 6.3 for an example of the specific situation that would be selected as the most similar to the current one.

CASE	DIST	I-Zn	O-SS	O-COD	I-SS	I-COD	I-BOD	Q	D-SS	D-COD	I-pH	O-pH
NEW		0.7	131	200	230	478	189	42393	104	412	7.9	7.6
5	0.019	1.0	74	200	222	495	186	NIL	112	NIL	7.6	7.6
20	0.020	2.0	60	110	240	500	410	28000	130	340	7.8	7.5
4	0.072	1.5	200	200	292	319	238	42857	104	276	7.7	7.4
19	0.074	0.7	104	172	116	252	114	42911	70	216	7.6	7.5
12	0.267	0.71	44	43	146	276	156	34573	124	176	7.7	7.7
1	0.384	1.5	21	84	166	407	NIL	44101	94	280	7.8	7.3

Table. 6.4. An example of most similar specific situation

Thus, the two diagnostics must be combined again, to obtain the most plausible situation. If both diagnostics are coherent, the degree of confidence on DAI-DEPUR competence increases. But it can happens that the two diagnosis are contradictory. Then, one heuristic rule could be to give higher priority to the experiential situation diagnosis, due to the fact that it is more specific (related to a concrete WWTP under supervision), than the generic situation diagnosed (related to general knowledge about the domain). But taking into account the environmental problem that DAI-DEPUR is dealing with, and the possible damage caused by wrong decisions, we decided that the operator has the ultimate choice to finally identify what he/she thinks that it is the most reasonable working situation, based on his/her experience. The operator can inquire DAI-DEPUR to get some data values, ask for some explanations of the deductive inferences, *etc.*, to confirm his/her opinions.

The *Explanation module* gives some explanations about the reached conclusions of the different KBS agents of the system (Screen KBS, Thickening KBS, *etc.*) and it can give some required reports about deductive processes.

Chapter 7

The Plans Level

The mind is the seed of the action.
Ralph Waldo Emerson

7.1 Introduction

The top level of the architecture, the plans level, manages different domain models: the situation finally identified in the WWTP, the first proposed solution, and the final adopted solution. The solutions are plans that can be derived from some actuation rules encoded in the Supervisory-KBS, from the solution given by the most similar specific situation in the CBRL agent memory, or from the numerical control algorithm if the *normal situation* has been diagnosed. These plans are a sequence of actions that must be taken to restore or maintain the correct WWTP operation and performance.

The main tasks involved at this level are the validation task and the actuation task. The validation task of the identified situation and the first proposed plan is achieved by the operator's validation method. The final actuation task over the WWTP operation is carried out by the expert actuation, experiential actuation, and the numerical control actuation method.

7.2 Domain Models

The two domain models used at this level are the final identified situation and the proposed and adopted solutions (plans).

7.2.1 Identified situation

The final identified situation was obtained in the previous level as a combination of the local situations into the generic situation diagnosed, and also taking into account the most similar specific situation. This identified situation can be also validated by the operator through the explanation module: inquiring for some variable values, analyzing some deductive inferences, etc.

The final identified situations can be one of the following generic situations:

- Foaming
- Rising
- Low dissolved oxygen (DO) Bulking
- Low Sludge Retention Time (SRT) Bulking
- Nutrient deficiency Bulking
- Sulphur Bulking
- F/M Bulking
- Non mechanical primary settlers problems
- Bioreactor anaerobic conditions
- Mechanical primary settlers problems
- Low primary settlers wasting
- Bad primary settlers wasting temporization
- Bad closed by-pass
- Anomalous pH or DO sensor signalling
- Turbines crash
- Broken clarifier bridge rotatory band
- Electrical power off
- Abnormal pH
- Abnormal environmental temperature
- Anomalous sludge retention time (SRT)
- Anomalous F/M ratio
- High secondary settlers sludge level
- Grease inflow
- Inflow sulphurs
- Toxic shock

- Loading shock
- Solids shock
- Storm
- Overflow

or a specific situation described by means of the values of some variables, as for example:

- ((Q 29719) (I-Zn 0.2) (I-pH 7.6) (I-BOD 133) (I-COD 284) (I-SS 186) (D-COD 160) (D-SS 52) (O-pH 7.8) (O-COD 60) (O-SS 21) …)

7.2.2 Proposed and adopted plans

A first plan (solution) is provided by extracting it either from the encoded rules embodied in the Supervisory-KBS or from the adapted solution of the most similar specific situation.

The main expert general actuation plans for the generic situations are:

Rising
ACTUATION: try to decrease, if possible, the sludge age to wash out nitrifying population, and reduce the dissolved oxygen levels in the bioreactors.

ACTUATION PLAN:
(Increase the Waste-flow) ;;; diminish the sludge age
(Decrease the Dissolved Oxygen in the bioreactor)

Foaming
ACTUATION: Foaming control by reducing the sludge age has long been employed in practice. In general, very low values of the sludge age (below three days) are required for this control method to be effective. Furthermore, you must be really careful with sidestreams to avoid a constant reinoculation of the filamentous organisms.

ACTUATION PLAN:
(Increase the Waste-flow) ;;; diminish the Sludge Retention Time
(Chlorinate foaming-areas)

Bulking

ACTUATION: Use the microscopic examination procedure to identify the causative filamentous organism. Combining this identification and manuals information, together with a knowledge of the plant operating conditions and wastewater characteristics, it is possible to determine the probable cause(s) of the filamentous organism(s) growth.

Some of the causes can be rectified with operational changes. If **septic water** is indicated, wastewater prechlorination may be initiated. If the probable cause **is nutrient deficiency**, determine which nutrient(s) is deficient by analysis of influent and effluent, and rectify the deficiency by increasing the feed rate of existing nutrient supply system(s) or by installing nutrient addition facilities.

Some plants with bulking sludge problems may require major design or operational changes that can take a long time to implement (e.g., additional **aeration** capacity, changes in aeration basin configuration, **industrial** waste control, decreasing the **SRT**, *etc*.). In addition, once changes have been made to discourage the growth of filamentous organisms, sludge settleability may improve really slowly.

Rapid, nonspecific methods to eliminate the symptoms of bulking fall into three categories:

a) manipulation of recirculation activated sludge (**RAS**) flow rates and wastewater feed points to the aeration basin.
b) addition of **chemicals** to enhance the settling rate of the activated sludge without attempting to selectively limit the growth of filamentous organisms.
c) addition of **toxicants** to the activated sludge to selectively kill the extended filamentous organisms that cause the bulking [Jenkins *et al.*, 1993].

Non-mechanical problems in primary settlers

ACTUATION: The proposed solution to a non mechanical problem in primary settler is always probable cause dependent. If there is floating sludge, you must remove sludge more frequently or at a higher rate. When the observation indicates black and odorous septic wastewater or sludge, the proposed solution is to increase frequency and duration of pumping cycles until sludge density decreases to an undesirable low value; you can also add chemicals or aerate in collector systems. If there is undesirable low solids contents in sludge, then reduce frequency and duration of pumping cycles, or provide more even flow distribution in all tanks, if multiple tanks. Finally, when suspended solids efficiency removal is poor, then try to use available tankage, shave peak flow or add chemicals [WPCF, 1990].

Bioreactor anaerobic conditions
ACTUATION PLAN:
(Increase the DO level)

Sidestreams
ACTUATION PLAN:
(Keep sidestreams flow in an empty tank)

Abnormal pH
ACTUATION PLAN:
(Add alkali or acid to neutralize the water)

Abnormal Environmental Temperature
ACTUATION PLAN:
(Modify the F/M ratio) ;;; to adjust the biomass level

Anomalous sludge retention time (SRT)
ACTUATION PLAN:
(Modify the waste-flow) ;;; to adequate the sludge age

Anomalous F/M ratio
ACTUATION PLAN:
(Modify the F/M ratio) ;;; to adjust the biomass level

High secondary settlers sludge level
ACTUATION PLAN:
(Increase the recirculation-flow)

Grease inflow
ACTUATION PLAN:
(Optimize the grit removal operation) ;;; to avoid grease entering to the bioreactor

Inflow sulphurs

ACTUATION PLAN:

(Add chemicals to remove inflow sulphurs)

Toxic shock

ACTUATION: It is necessary to determine the kind and the quantity of heavy metal content in the influent. If the level is significant the influent must be rejected, and the source (an industry) must be identified. This metal could be accumulated in the activated sludge due to adsorption process, so, it will be necessary to increase de Waste Activated Sludge (WAS) flow to remove the metal of the process.

ACTUATION PLAN:

(Increase the Waste-flow) ;;; avoiding toxic accumulation

Loading shock

ACTUATION PLAN:

(Increase the Recirculation-flow) ;;; to restore F/M ratio balance

Solids shock

ACTUATION PLAN:

(Increase the frequency of the automatic cleaning grids of the preatreatment)
(Increase the frequency of the grit removal bridge)
(Increase the sludge purge in the primary settlers)
(Take care of the sidestreams)

Storm

ACTUATION: When the storm is really important, a fraction of the influent may by-pass the plant to prevent a wash-out of the microorganism population and an hydraulic shock in the clarifiers.

ACTUATION PLAN:

(Decrease the Recirculation-flow)
(Open the by-pass if necessary)
(Use when available an equalization tank)

Overflow

ACTUATION PLAN:

(Decrease the Recirculation-flow)
(Open the by-pass)
(Use when available an equalization tank)

The experiential specific actuation plans are based on the previous actuation plans of other similar specific situations occurred in the WWTP. As an example, the actuation plans for the initial set of cases of the Case library are detailed in table 7.1.

CASE	DAY	SITUATION	ACTUATION PLAN
C-1	1/3/90	HIGH LOADING	INCREASE RECIRCULATION-FLOW 10 %
C-2	29/6/90	NORMAL	
C-3	13/3/90	POSSIBLE BULKING	CONFIRM BULKING IDENTIFY KIND OF BULKING
C-4	14/3/90	BULKING	IDENTIFY KIND OF BULKING ACTUATE OVER KIND OF BULKING
C-5	18/7/91	POSSIBLE BULKING	CONFIRM BULKING IDENTIFY KIND OF BULKING
C-6	1/10/90	STORM	DECREASE RECIRCULATION-FLOW 20 %
C-7	31/3/91	NORMAL	
C-8	31/5/91	SOLIDS SHOCK	INCREASE PRETREATMENT-EFFICIENCY
C-9	5/6/90	SOLIDS SHOCK	INCREASE PRETREATMENT-EFFICIENCY
C-10	29/4/90	POSSIBLE BULKING	CONFIRM BULKING IDENTIFY KIND OF BULKING
C-11	14/9/90	RAIN	DECREASE RECIRCULATION-FLOW 15 %
C-12	1/4/91	NORMAL	
C-13	5/8/91	LOW LOADING	DECREASE RECIRCULATION-FLOW 10 %
C-14	12/8/90	STORM	DECREASE RECIRCULATION-FLOW 30 %
C-15	1/2/91	HIGH LOADING	INCREASE RECIRCULATION-FLOW 10 %
C-16	23/11/90	LOW LOADING	DECREASE RECIRCULATION-FLOW 10 %
C-17	22/10/90	STORM	DECREASE RECIRCULATION-FLOW 30 %
C-18	24/5/91	SOLIDS SHOCK	INCREASE PRETREATMENT-EFFICIENCY
C-19	15/3/90	POSSIBLE BULKING	CONFIRM BULKING IDENTIFY KIND OF BULKING

Table 7.1 Specific actuation plans for the initial Case Library

7.3 Tasks

Plan validation and actuation are the tasks involved within this level. The goal of both tasks is to ensure a reliable and safe actuation over the WWTP.

7.3.1 Plan Validation

The WWTP operator must validate the proposed actuation plan. DAI-DEPUR shows him/her through the user interface, and the operator can reject the plan completely, or modify some parts of it. When the operator has arranged the plan accordingly with its experience and idiosyncratic point of view, then, DAI-DEPUR is provided with the adopted plan that must be translated in a next step into a set of operational actions.

7.3.2 Actuation

If the WWTP's operation is correct, the activated sludge process can achieve high depuration levels (plus than 90 % of COD removal) and cope with loading shocks and other perturbations. Operation is an important feature because it is through itself that versatile process conditions and process complexity can be managed.

A WWTP operation providing high depuration levels requires a set of actions to adequate at each step, the parameters of the process to the changing conditions of the process, so that WWTP never are in steady-state conditions. Thus, the actuation task is aimed to actuate over the WWTP. Direct actuation is possible if there are on-line actuators to modify the main operating parameters of the WWTP: turbines on/off (aeration), recirculation-flow and waste-flow. If not, manual actuation off the WWTP staff is required.

If the diagnosed working situation of the plant is *normal*, then the automatic numerical control is activated/maintained, to continue schedulling the aeration motor switches of the bioreactor. Otherwise (*abnormal situation*) experiential/expert actuation is done based on general actuation plans (predefined plans) or specific actuation plans (experienced plans), always under the supervision of the operator.

7.4 Methods

The methods that implement the plan validation task and the actuation task are: the operator's validation, the expert actuation, the experiential actuation and the numerical control actuation.

7.4.1 Operator's Validation

The operators must validate the suggested actuation plan to ensure a safe and reliable WWTP operation. To this end, the operator can inquire DAI-DEPUR about some variables' values, to ask for some information about the reasoning processes followed by the system, or to start same simulation proofs. All these validation features can be achieved through the explanation and inspection abilities supported by the G2 shell, and visualized by means of the user interface, as shown in the chapter 9.

7.4.2 Expert/Experiential Actuation

Expert actuation is implemented with some actuation rules provided by the experts and the Chemical Engineering literature to cope with *usual abnormal situations* in the Supervisory-KBS. See figure 7.1 for an example of actuation rules.

Experiential actuation is performed by some actuation plans stored in the Case Library and experienced in the WWTP under supervision, to cope with *unusual abnormal situations*. These actuations plans are previously adapted to perfectly match the current WWTP situation.

Sometimes, the actuation plan is a mixture of both suggested plans. The operator can combine and modify both proposed plans and his/her own plan into the finally adopted actuation plan.

SUPERVISORY-KBS-RULE-007 ::
IF SIDESTREAMS situation is VERY-
THEN
Keep sidestreams flow in an empty tank AND
CONCLUDE THAT
Kept-Sidestreams situation is TRUE

SUPERVISORY-KBS-RULE-022 ::
IF Bioreactor-pH-sensor is bad ALMOST-SURE
THEN
Calibrate Bioreactor-pH-sensor AND
Add alkali to neutralize the mixed-liquor

SUPERVISORY-KBS-RULE-023 ::
IF wasting-pipe is plugged VERY POSSIBLE
THEN
Send the operator to unplug the wasting-pipe

SUPERVISORY-KBS-RULE-018 ::
IF Fear-of-bulking is TRUE
THEN
Identify the causative filamentous organism

SUPERVISORY-KBS-RULE-019 ::
IF STORM situation is VERY POSSIBLE
THEN
Decrease the Recirculation-flow AND
Open the by-pass if necessary AND
Use when available an equalization tank

Fig. 7.1 A set of actuation rules

7.4.3 Numerical Control Actuation

Numerical control actuation is implemented by a predictive DO control algorithm (explained in 5.2.1) that is suitable to be applied when the WWTP is operating in normal conditions (*normal situations*). This algorithm provides DAI-DEPUR with a schedulling of aeration motor switches in next time horizon.

Chapter 8

Experimental Evaluation and Validation

Después de la verdad, nada hay tan bello como la ficción.
Antonio Machado

8.1 Introduction

The evaluation of DAI-DEPUR is not an easy task. First, a direct evaluation of the system in a real WWTP cannot be carried out, due to the great damage thay can be caused to the environment. Secondly, DAI-DEPUR is composed of several KBS, the CBRL agent, the supervisory-KBS, the numerical control knowledge module, *etc.*, and so, a detailed validation of each main components will be needed. To achieve a good evaluation of DAI-DEPUR, it was decided to perform the validation in two steps:

- Validation of each main component: evaluation of the numerical control knowledge module, evaluation of the expert knowledge paradigm and evaluation of the experiential knowledge paradigm.

- Validation of the whole system. The system is being validated at three levels:

  a) Simulation of the WWTP in real-time with expert's validation
  b) Building-up and testing on a pilot scale WWTP
  c) Validation on a real WWTP

8.2 Single Validation of the Components

8.2.1 Numerical Control Knowledge Validation

The numerical DO control algorithm, described in 5.2.1, has been applied in simulation using experimental data obtained from Manresa's WWTP [Moreno *et al.*, 1992; Moreno, 1991]. The objective of this WWTP is to try to keep always the DO level over 2 mg/L, to allow the substrate oxidation reactions to proceed normally. In order to attain it, in despite of inflow and load variations, the sequence of motor switches is defined by the plant manager, using his own experience about the plant dynamics. The sequence is established for two days, and re-evaluated after this time according to the results obtained.

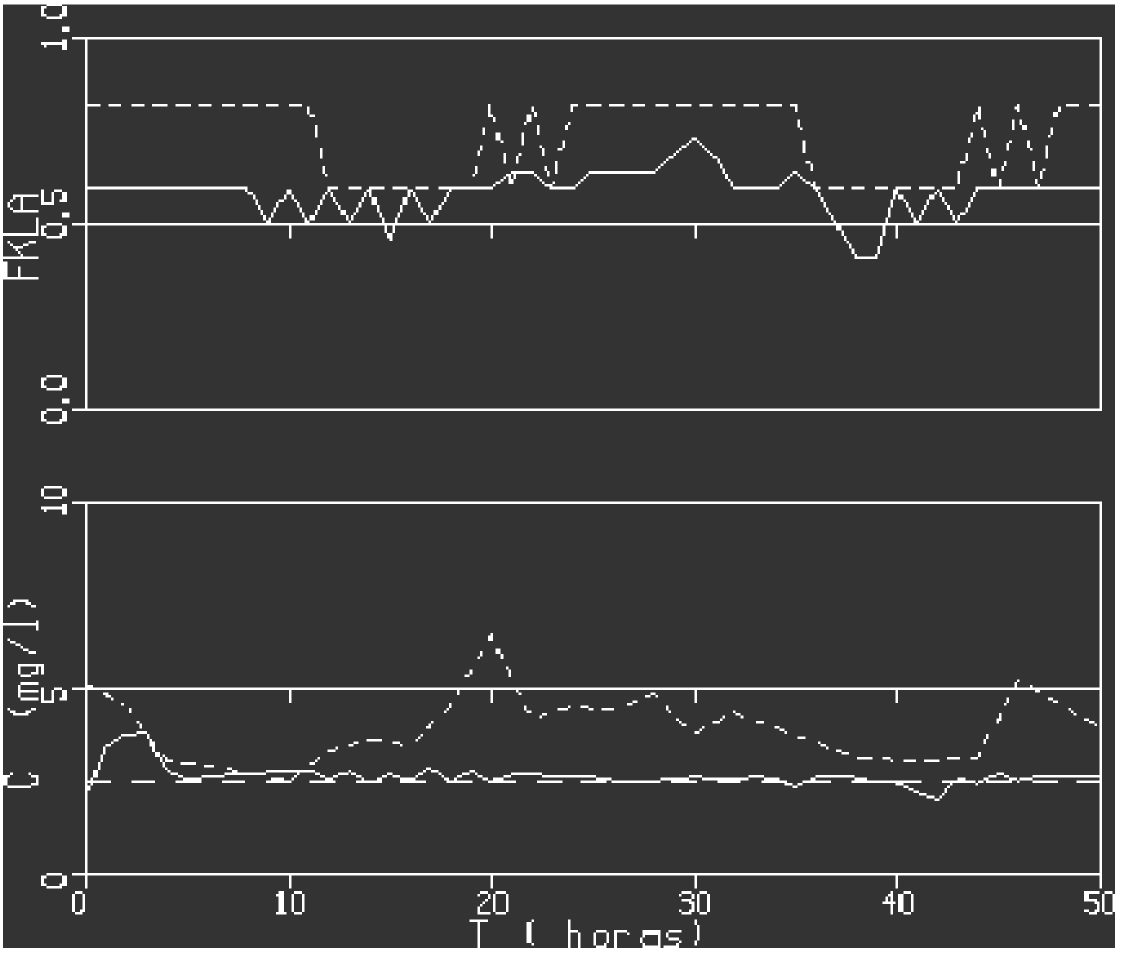


Fig. 8.1 Control algorithm results (a) Control actions and (b) DO concentrations

In figure 8.1 (a), the real power factor profile used along two days in the plant is presented (dashed graphic), together with the obtained using the proposed non-linear DO predictive control algorithm (continuous graphic). As it may be observed, the use of the proposed strategy allows to obtain a significative reduction of energy consumption (in this case evaluated about a 35 %).

The DO evolution, experimental and controlled by the proposed algorithm is presented in figure 8.1(b). As it is shown in the figure 8.1, for both cases, the DO level is always greater than 2 mg/L, as required. But, there is a period of time in which experimental DO concentration (dashed graphic) is greater than 4 mg/L, a situation in which part of the energy consumed is not efficiently used. Figure 8.1 (b) also describes the DO evolution obtained using the motor switches presented in figure 8.1 (a), computed by the proposed control method (continuous graphic). In this case, selecting a set point of 2.5 mg/L, the DO values are very close to the set point, allowing the algorithm to maintain the required DO level in a more efficient way, and always satisfying the imposed constraints.

The results presented in this evaluation have been obtained with a sampling period of 30 minutes, and prediction and control horizons of PH = 10 and CH = 3 sampling periods, respectively.

Thus, the non-linear DO predictive control methodology, seems to be an efficient way to control the DO levels in the activated sludge process. It allows to maintain always the DO level over the required value of 2 mg/L with a significative reduction of the energy used in the aeration process. In addition, the algorithm has shown to be able to successfully solve the problem of managing the technical constraints present in these plants, without a significative loss of performance.

8.2.2 Expert Knowledge Validation

In order to validate the knowledge acquisition tool –LINNEO$^+$–used to build-up the knowledge bases of the different Knowledge-Based Systems that integrate DAI-DEPUR, was made a compared study of LINNEO$^+$ against a well known classification tool, the K-means method. The results given in this chapter correspond to a study carried out in the Manresa's WWTP [Sànchez *et al.*, 1995e; De Gràcia, 1993]. Also, the knowledge bases obtained from the knowledge acquisition module were tested against the experts' opinion with very good results.

The use of unsupervised learning techniques for the (semi) automatic creation of Knowledge-Based Systems (KBS) has been an important research activity in the AI community. Our knowledge acquisition tool, LINNEO$^+$, tries to overcome some classical problems in the field such as order sensitivity, lack of stability, *etc.* [Weinberg *et al.*, 1992].

LINNEO$^+$ have been tested with success against other systems such as COBWEB [Fisher, 1987] and K-means [Dubes and Jain, 1990] (see [Béjar, 1995]). The comparison with K-means is chosen because our experts had long been working with that system and its comparison with LINNEO$^+$ does not need further data manipulation. Only the K-means method will be presented, because LINNEO$^+$ methodology was already described in 5.4.1.1. It is important to notice that only after the experts used LINNEO$^+$ for the first time, they think on give names or identify situations using K-means or other clustering method.

Variables	Situation at the plant
Q ZN PH BOD COD SS SSV SED COND	Input plant
PH BOD SS SSV SED COND	Primary treatment input
PH BOD COD SS SSV SED COND	Secondary treatment input
PH BOD COD SS SSV SED COND	Output plant
RD_BOD RD_SS RD_SED	Efficiency primary treatment
RD_BOD RD_COD	Efficiency secondary treatment
RD_BOD RD_COD RD_SS RD_SED	Overall efficiency

Symbol	Description	Units
Q	Flow	m^3/day
Zn	Concentration of zinc	mg/L
BOD	Measure of the biodegradable organic matter	mg/L
COD	Measure of the chemical oxidable organic matter	mg/L
SS	Measure of suspended solids	mg/L
SSV	Volatile suspended solids. This variable is related with the biomass in the sample	mg/L
SED	Measureof the sedimentable solids	mg/L
COND	Electric conductivity. This variable is related with dissolved salts in the sample.	mg/L

Table 8.1. List of experimental variables considered in the classification study

8.2.2.1 The Data Stream

The data studied correspond to data collected on the operation of the Manresa's WWTP during the period 1990-1991. A total of 527 sets of data have been considered, each of which consists of 38 variables. Of these 38 variables, 29 correspond to measurements taken at different points of the plant, while the remaining 9 variables correspond to the calculated performance of the primary and secondary treatments for the entire plant.

A complete list of the variables is given in Table 8.1, together with a brief comment on the meaning of each variable. These data are available by anonymous ftp from the UCI Machine Learning Repository of data bases.

8.2.2.2 K-means Method

The purpose of K-means [Dubes and Jain, 1990] is to classify a set of data in such a way that those belonging to a given group are as similar as possible. To this end, the matrix of similarity must be established. Each element of the matrix has a number which is the measure of similarity between each pair of objects. Of the various existing techniques for constructing groups, the one used in this case has been the K-means method, based on division techniques, and which uses Euclidean distance. This method requires an *a priori* definition of a certain number of groups. Analysis rearranges the objects on the basis of the variables selected, so that at the end of the process they are as similar as possible. To solve the problem of an *a priori* definition of the number of categories, it is advisable to repeat the classification process with a different number of groups. It is an incremental classification method.

At each step, a new data distribution within classes is built, in order to optimize some predefined criterion. The criterion used was the maximization of the next statistic:

$$F = |A| \ / \ |W|$$

where $|A|$ is the determinant of the inter-groups covariance matrix and $|W|$ is the determinant of the intra-groups covariance matrix.

In the present study it has been used the implementation of these algorithms on *Systat* software.

8.2.2.3 Results obtained with LINNEO$^+$

First, the data (x_{ij}) were standardized (x_{ij}'), using a scaling transformation over the interval [0,1]:

$$x_{ij}' = (x_{ij} - x_{min}) \;/\; (x_{max} - x_{min})$$

x_{min} and x_{max} being the minimum and maximum values, respectively, for the variable under consideration.

In this case, the classification algorithm begins by selecting the best class among the current ones, for each of the objects in the data set. These classes are identified by experts as *situations*. The best for an object is the one in the previous set of classes with minimum distance from the object.

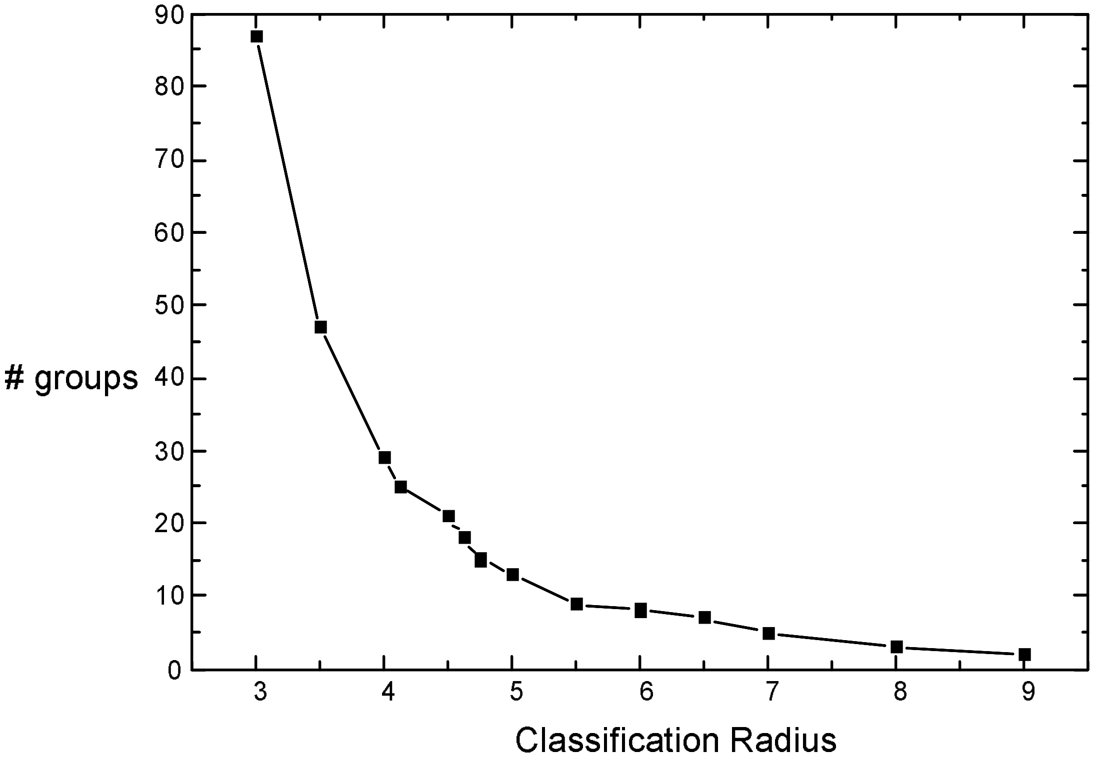


Fig 8.2. Variation of the number of classes according with the radius using LINNEO$^+$

Unlike the cluster analysis where the number of classes is fixed *a priori*, in this case the selection of the number of classes is carried by means of the classification radius. The number of classes obtained by LINNEO$^+$ is a function of the radius. If the radius is very small, the number of classes increases, and as the radius increases the number of classes decreases. Figure 8.2 shows the variation of the number of classes with the radius for the case studied. To compare the LINNEO$^+$ results with those of the K-means method, it was decided to work with 13 classes, the chosen radius therefore being 5.

Table 8.2 gives an example of the information for a *situation*, in this case no. 11, provided by LINNEO$^+$. Thus, LINNEO$^+$ indicates the number of days constituting the class, and the coordinates of the class centre in the space of 38 dimensions. In this case, a value of zero indicates that the variable has a minimum value, while the maximum value corresponds to 1.

"name: "#: | Class-0-11 |	
"components: " D-4/4/91 D-6/5/91 D-31/1/91 D-30/1/91 D-29/1/91 D-28/1/91 D-27/1/91 D-24/1/91 D-23/1/91 D-22/1/91 D-21/1/91 D-7/1/91 D-4/1/91 D-3/1/91 D-28/2/91 D-27/2/91 D-26/2/91 D-25/2/91 D-19/2/91 D-15/2/91 D-13/2/91 D-12/2/91 D-11/2/91 D-10/2/91	
D-7/2/91 D-6/2/91 D-4/2/91 D-1/2/91 D-22/3/91 D-21/3/91 D-19/3/91 D-15/3/91 D-14/3/91 D-13/3/91 D-19/10/90 D-30/11/90 D-29/11/90 D-28/11/90 D-26/11/90 D-25/11/90	
D-23/11/90 D-22/11/90 D-15/11/90 D-26/12/90 D-21/12/90 D-14/12/90 D-12/12/90 D-12/12/90 D-11/12/90 D-10/12/90 D-6/12/90 D-4/12/90 D-2/12/90	
"center: "	
Q-E: 0.45387453457474364	ZN-E: 0.08062930742289007
PH-E: 0.6823899371069181	BOD-E: 0.5327523063372119
COD-E: 0.5173102237823607	SS-E: 0.06152326385458858
SSV-E: 0.8207021600882953	SED-E: 0.14420587735975746
COND-E: 0.46532589053823676	PH-P: 0.6965408805031446
BOD-P: 0.5231439324798386	SS-P: 0.08773347274369087
SSV-P: 0.7247772536687632	SED-P: 0.1076592020205809
COND-P: 0.48849535029752117	PH-D: 0.7561683599419449
BOD-D: 0.5214794709255332	COD-D: 0.6412905485269013
SS-D: 0.30817610062893075	SSV-D: 0.7337683832222064
SED-D: 0.2174310842881841	COND-D: 0.49392615078638164
PH-S: 0.30125706451124357	BOD-S 0.05267472741135026
COD-S: 0.23072451095247104	SS-S: 0.05198728303382467
SSV_S: 0.7657508606256699	SED-S: 0.005073975444149756
COND-S: 0.351599571994931	RD-BOD-P: 0.5241987819392059
RD-SS-P: 0.525479747745348	RD-SED-P: 0.8411583638259165
RD-BOD-S: 0.9165440284791425	RD-COD-S: 0.7708890987456679
RD-BOD-G: 0.9311167888553367	RD-SS-G: 0.9079742460821214
RD-SED-G: 0.9917660974517388	

Table 8.2. An example of output of a class provided by LINNEO$^+$

As in the case of K-means, the classes were interpreted as a function of their centres, giving the results presented in Table 8.3.

Cluster #	Days	Operation	Characterization	Classification
1	275	Right		Normal
2	1 (13/3/90)	Out of limits	Operation	Problems with secondary treatment
3	1 (14/3/90)	Out of limits	Operation	Problems with secondary treatment
4	4 (15/3/90-17-18-19/7/91)	Out of limits	Operation	Problems with secondary treatment
5	116	Right		Normal
6	3 (5/6/90-28-31/5/91)	Right	Input	Overloading
7	1 (29/4/90)	Out of limits	Operation	Problems with primary and secondary
8	1 (14/9/90)	Right	Input	Storm
9	69	Out of limits		Normal
10	1 (12/8/90)	Right	Input	Storm
11	53	Right		Normal
12	1 (22/10/90)	Right	Input	Storm
13	1 (24/5/91)	Right	Input	Overloading

Table 8.3. List of classes obtained by LINNEO$^+$ and expert's interpretation

8.2.2.4 Results obtained with K-means Analysis

First, the data were scaled down so that the standardized variables had a mean of zero and a variance of 1, using the formula:

$$z_{ij} = (x_{ij} - xm_j) / s_j$$

x_{ij} being the value to be scaled, xm_j the mean value of the set j, and s_j the standard deviation of the set j.

Two parallel processes are then carried out, a *cluster analysis* to classify the data, and a *principal components analysis* which will enable us to visualize the results and will assist in the classification process.

Bearing in mind that it is necessary to make an *a priori* definition of the number of classes, it was decided, after carrying out some tests, to perform the study with 13 classes. It was felt that this number allowed the greatest possible number of *situations* to be considered, without being excessive. From the application of the algorithm, the characteristics of the 13 classes defined by the distance of the data from the centre of the class were obtained, together with the degree of variation and the standard deviation of the variables forming the centre of the class.

CLUSTER NUMBER: 4

MEMBERS		STATISTICS				
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
15/3/90	1.55	Q_E	-0.89	-.3838E-02	0.87	0.56
27/4/90	1.40	ZN_E	-0.61	-0.35	0.23	0.32
17/7/91	1.01	PH_E	-3.70	-1.26	-0.04	1.26
18/7/91	0.96	BOD_E	-1.23	-0.08	0.80	0.78
19/7/91	0.98	COD_E	-1.29	0.15	1.19	0.89
		SS_E	-0.82	-0.12	0.11	0.36
		SSV_E	-0.23	0.25	0.80	0.35
		SED_E	-1.27	-0.24	0.34	0.56
		COND_E	-0.75	-0.38	0.10	0.30
		PH_P	-2.34	-0.66	0.31	0.98
		BOD_P	-0.81	-0.04	1.11	0.67
		SS_P	-0.80	-0.16	0.18	0.36
		SSV_P	-0.15	0.25	0.55	0.23
		SED_P	-0.62	0.11	0.91	0.56
		COND_P	-0.82	-0.50	-.8484E-04	0.30
		PH_D	-2.56	-0.96	-0.06	0.86
		BOD_D	-1.20	-.9680E-02	1.49	0.87
		COD_D	-0.79	0.51	1.93	0.94
		SS_D	-1.01	0.64	1.83	0.92
		SSV_D	-1.03	-0.29	0.96	0.68
		SED_D	-0.31	0.93	2.65	1.09
		COND_D	-0.78	-0.40	0.04	0.33
		PH_S	-1.12	-0.48	0.48	0.62
		BOD_S	1.11	3.84	4.94	1.43
		COD_S	2.21	4.38	5.34	1.22
		SS_S	3.18	4.09	5.03	0.79
		SSV_S	-0.64	0.10	0.70	0.46
		SED_S	-0.18	0.44	1.37	0.60
		COND_S	-0.71	-0.17	0.38	0.35
		RD_BOD_P	-0.33	0.07	0.54	0.36
		RD_SS_P	-0.79	-0.52	0.12	0.32
		RD_SED_P	-2.49	-0.66	0.58	1.14
		RD_BOD_S	-8.89	-5.11	-.5381E-05	3.92
		RD_COD_S	-5.71	-3.17	.2907E-04	1.86
		RD_BOD_G	-9.25	-6.00	-0.81	3.06
		RD_COD_G	-6.76	-4.60	-3.21	1.31
		RD_SS_G	-9.64	-4.38	-2.59	2.67
		RD_SED_G	-0.95	-0.46	0.21	0.51

Table 8.4. An example of output of a class provided by K-means

As an example, the *situation* corresponding to class 4 is presented (see Table 8.4). The variables are scaled so that a positive value indicates that the measurement is above the mean, and the greater this value is, the further away from the mean it will be. Thus, in this class it can be observed that the value of the output BOD (BOD_S) has deviated 3.84 standard deviation units from the mean. This indicates that one of the characteristics of this class will be a high output BOD.

Table 8.5 is obtained from the study carried out by the experts for each of the characteristics of the classes. For each class, it presents the number of days of which it consists (in the case that the class comprises few days, the date of each one is given) and the interpretation of how the plant has operated in that *situation*. The conclusion is that the plant functions correctly on most of the days concerned, notwithstanding different operating *situations* and inflow characteristics. The classification process and a study of the classes obtained also make it possible to identify the alteration which has led to the malfunction of the plant.

Cluster #	Days	Operation	Characterization	Classification
1	175	Right		Normal
2	1 (13/3/90)	Out of limits	Operation	Problems with secondary treatment
3	181	Right		Normal
4	5 (15/3/90- 27/4/90 17-18-19/7/90)	Out of limits	Operation	Problems with secondary treatment
5	123	Right		Normal
6	2 (5/6/90- 28/5/91)	Right	Input	Overloading with suspended solids
7	1 (24/5/91)	Right	Input	Overloading without suspended solids
8	23	Right	Operation	Problems with primary treatment
9	1 (29/4/90)	Out of limits	Operation	Problems with primary and secondary

10	1 (12/8/90)	Right	Input	Storm
11	2 (8/7/91-9/7/91)	Right	Input	Toxic loading
12	11	Right	Input	Overloading
13	1 (14/3/90)	Out of limits	Operation	Problems with secondary treatment

Table 8.5. List of classes obtained by K-means and expert's interpretation

Figure 8.3 shows the experimental data as a function of the first three principal components. From this figure, it can be seen that the differentiated *situations* correspond to a *solids shock* (classes 6 and 7) and *problems in the secondary clarifier* (classes 2, 9 and 13). In the case of the remaining groups obtained by means of cluster analysis, graphic differentiation has not been possible.

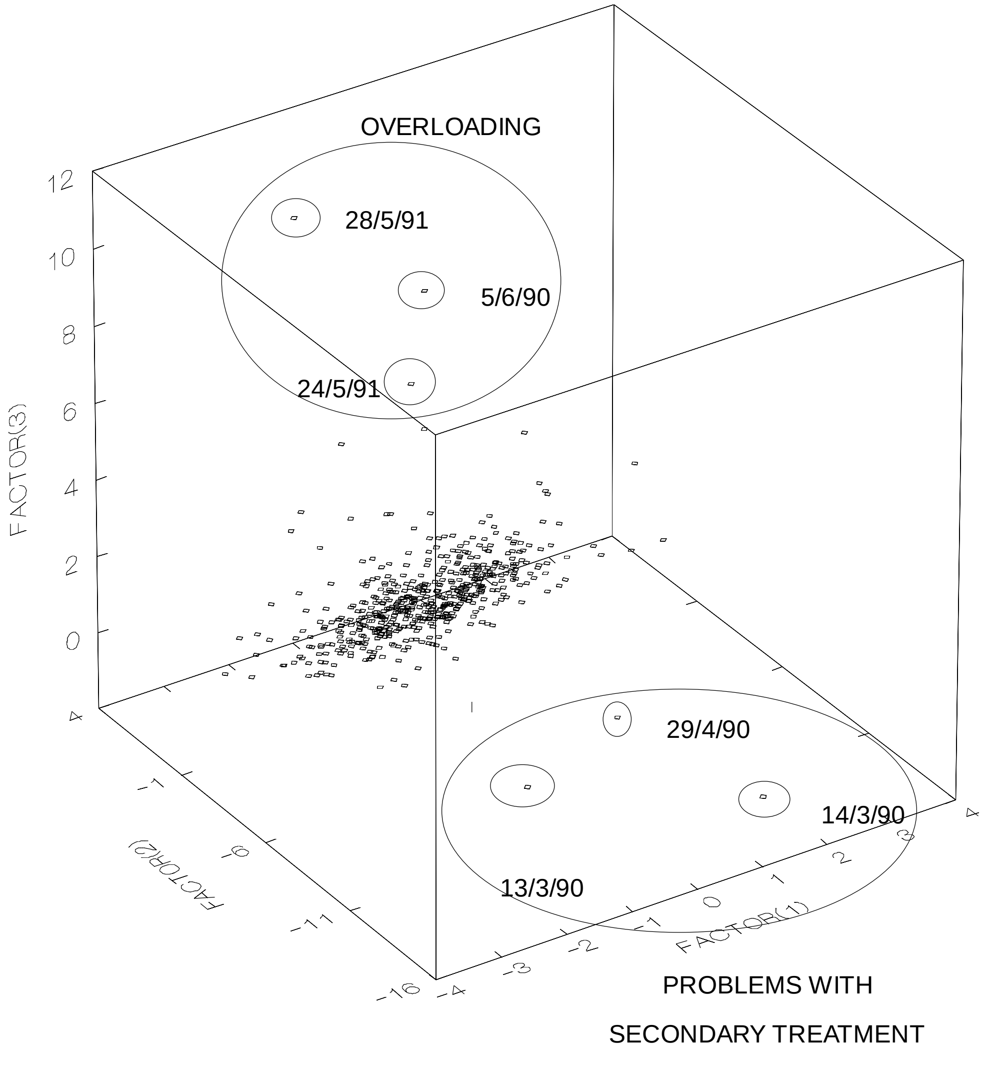


Fig. 8.3. Representation of experimental data as a function of first principal components

8.2.2.5. Comparison of Classification Results

Once the variables were defined, a preliminary study of the data matrix was carried out. This study has made it possible to discover the overall behaviour of the plant, as regards the characteristics of the inflow, cleansing percentages during each of the stages, and overall behaviour of the plant with regard to maintaining the required water quality limits.

The results of classification using the two methodologies were compared on the basis of the components of each of the classes and their interpretation as *situations*. The results obtained are shown in Table 8.6. As can be seen, there are two different kind of *situations*, that which corresponds to *operational problems*, and that which corresponds to the *normal behaviour* of the plant.

K-means			LINNEO+		
G. CLUSTER	N. days	Days	G. LINNEO+	N. days	Days
1	175		1	275	
2	1	13/3/90	2	1	13/3/90
3	181		11	53	
4	5	15/3/90 27/4/90 17-18-19/7/91	4	4	15/3/90 17/7/91 18-19/7/91
5	123		5	116	
6	2	5/6/90-28/5/91	6	3	5/6/90 28/5/91 31/5/91
7	1	24/5/91	13	1	24/5/91
8	23				
9	1	29/4/90	7	1	29/4/90
10	1	12/8/90	10	1	12/8/90
11	2	8-9/7/91			
12	11				
13	1	14/3/90	3	1	14/3/90

Table 8.6. Comparison of classes obtained using both classification methods

In the first case, there is a high degree of coincidence in the classification process. Thus, classes 2, 4, 6, 7, 9, 10 and 13 of the cluster analysis coincide almost exactly with classes 2, 4, 6, 13, 7, 10 and 3 of the LINNEO$^+$ analysis. This indicates that both methods allow a differentiated classification of the days on which there is some malfunction of the plant and, in particular, a grouping of those days on the basis of the problem occurring in each *situation*.

In the second case, the classes which group together the days considered as reflecting the normal working of the plant are almost equivalent, although they are distributed in different ways. The cluster analysis establishes three major groups of 175, 181 and 123 days which, despite some slight differences among them, are grouped by the experts under the heading of correct operating conditions. The analysis performed by LINNEO$^+$ establishes three classes, with a more differentiated number of days for each class. Thus, it gives one class with 275 days, another with 116, and finally another with 53. As in the previous case, although slight differentials can be found, the *situations* grouped together in each of the classes are very similar. But the interpretation is richer in the case of LINNEO$^+$ because it also provides a chance to interpret the *normal situations* more precisely. As explained in chapter 5 these results lead to the identification of 4 types of *normal situations* that were previously mixed in only one by the experts.

When this study is carried out over 25 classifications with different data orders, one can observe that although the similarity above described, the K-means method results are less stable, that is, more objects move from one class to other. A detailed study of this comparisons can be found in [Béjar, 1995]. Also, the intensional description of LINNEO$^+$ results are easier to interpret by the experts, so that they can identify the *situations* by its values in a very direct way.

8.2.2.6 Obtained Situations Versus *a priori* Defined Situations

After obtaining a set of possible operating conditions for the plant, it was considered useful to compare these *situations* –obtained on the basis of the experimental operating data–, with a set of *a priori* defined *situations* by the experts in a preliminary study. In that study [Serra *et al.*, 1994], the *situations* were obtained from a classification of the defined attributes of each of the objects, getting a total of 19 possible *situations*. The comparison carried out is given in Figure 8.4.

Some differences can be seen. Some of the *situations* in the plant identified by the two procedures are similar, as in the case of the *normal situation* and that which corresponds to a *storm*. A second instance corresponds to the *problems in the secondary clarifier*. All these problems are grouped into a single class in the classification process, it having proved impossible to discriminate what the specific cause had been. From conversations with the plant's responsible, it was concluded that the *situation* which arose during that period corresponded to a *break in the conveyor belt*, while the *bulking situations* defined by the experts **had not occurred** during the period of time under study.

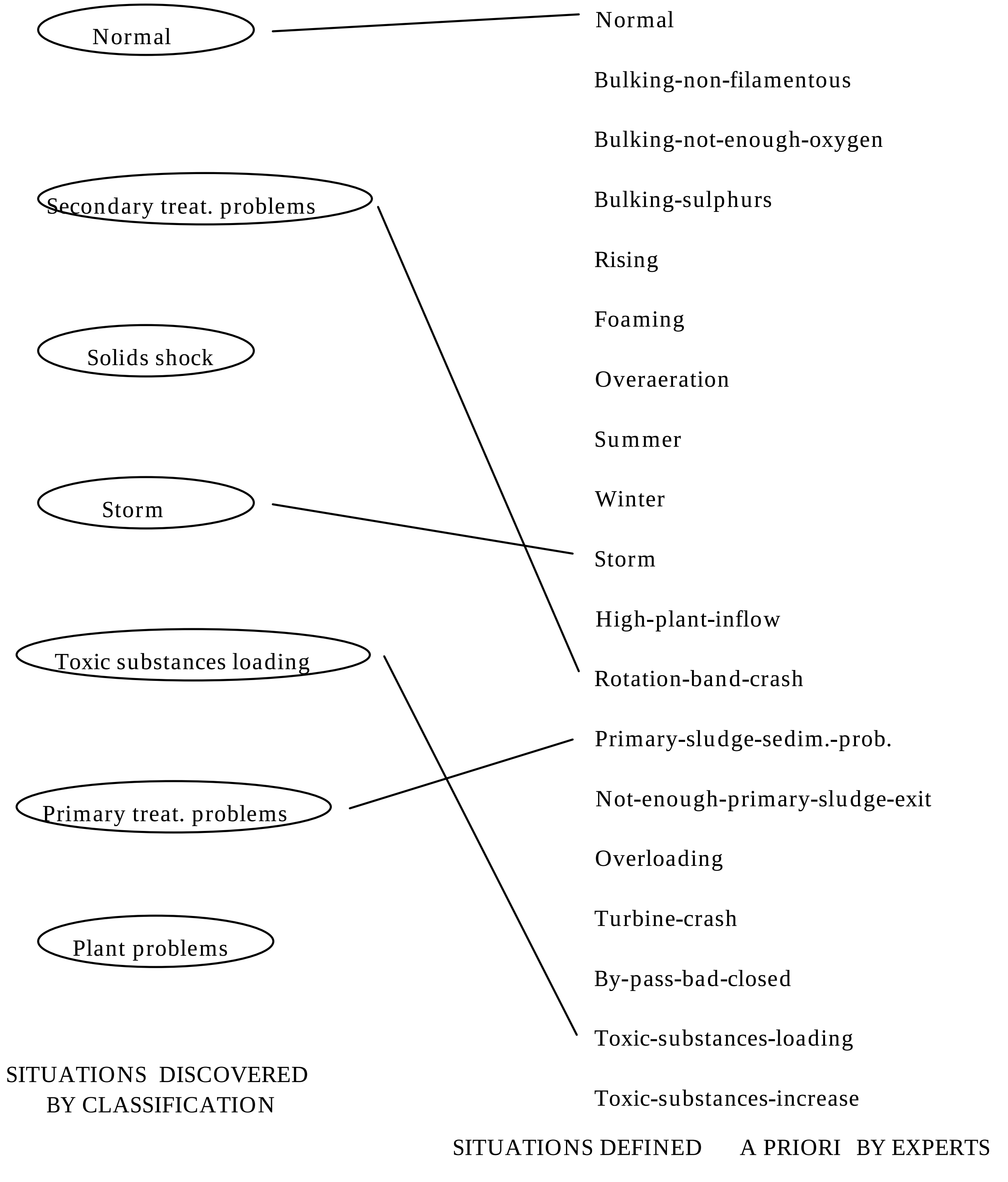


Fig. 8.4. Comparison of obtained classification situations versus *a priori* situations defined by experts

The third group appears to correspond to those classes (*situations*) identified in the classification process, but which had not been previously considered by the experts. In our opinion, this feature is of great interest because it indicates that the classification process (and its subsequent interpretation) provides information concerning the plant **not previously** considered by experts. On evaluating this fact with the experts, they agreed that it offers new perspectives on parameter generating processes.

The fourth group corresponds to *situations* defined by the experts which did not occur in the period of time considered (two years). This is perfectly possible, since these *situations* were obtained from *a priori* definitions of all the possible *situations*, which obviously were not necessarily going to occur during the period of time under consideration.

Finally, LINNEO$^+$ relevance methodology [Belanche and Cortés, 1991] had shown that a set of variables –those (9) corresponding to the calculated performance at several points of the plant– were *nought (do not care)*, that means that they do not contribute to concept construction and that can be eliminated from the data stream. This fact represents a great advantage over other traditional systems and led experts to re-interpret and re-formulate some part of their knowledge. In chapter 5, we had shown the importance of reducing the number of variables using some selection method.

8.2.3 Experiential Knowledge Validation

The Case-Based Reasoning (CBR) approach (experiential knowledge) has been validated taking into account different features: the similarity measure, the competence of the CBR, and the performance of the CBR. The case library is implemented as a prioritized discrimination tree, where the priority of node-attributes is obtained from experts' opinion and validated with an inductive learning method (ID3) [Segarra, 1995]. All CBR methods are designed and implemented, although we are refining the adaptation and evaluation steps. The case-based system is implemented in Common-Lisp.

8.2.3.1 The similarity measure

Our similarity measure, the *Eixample distance*, already described in 5.4.2.2 was defined after a wide performance study of different distances, mainly derived from the Minkowski metric:

$$d(x_i,x_j) = \left(\sum_{k=1}^{n} |x_{ik} - x_{jk}|^r\right)^{1/r} \qquad \forall\, r \geq 1$$

The study was carried out with the next distances, normalized over the interval [0,1]:

- Manhattan / City-block / Hamming distance: the Minkowski metric with r=1
- Euclidean distance: the Minkowski metric with r=2
- 3-order Minkowski distance: the Minkowski metric with r=3
- Exponential weighted Manhattan distance: a Manhattan distance with exponential weights

All the distances were tested in three variational ways depending on which value of the ordered attributes were used in the computation of the distance value: *discrete* or quantitative values, *continuous* or quantitative values, and *weight-sensitive* or discrete / continuous values depending on the weights of the attributes. Thus, the final comparison study was performed among the following possible distances:

- Discrete Manhattan distance
- Continuous Manhattan distance
- Weight-sensitive Manhattan distance
- Discrete Euclidean distance
- Continuous Euclidean distance
- Weight-sensitive Manhattan distance
- Discrete 3-order Minkowski distance
- Continuous 3-order Minkowski distance
- Weight-sensitive 3-order Minkowski distance
- Discrete Exponential-weighted Manhattan distance
- Continuous Exponential-weighted Manhattan distance
- Weight-sensitive Exponential-weighted Manhattan distance, i.e. *Eixample distance*

To evaluate the performance of the different distances, the case library was initialized with a representative set of cases obtained from a previous classification.

Then, 20 cases were used as a training set. For each one, the case library was searched to retrieve the most similar cases to it, using all the distances. The retrieval tables generated were studied by the experts, who marked each retrieval table for each possible distance, giving the following results:

- For each kind of distance, Manhattan, Euclidean, 3-order Minkowski, and Exponential-weighted Manhattan, always the best performing one is the weight-sensitive, afterwards the discrete one, and finally the worse is the continuous.

- Comparing the continuous distances, was discovered that all share the same degree of performance.

- The discrete distances all have an equal performance, excepting the Exponential-weighted Manhattan, that is the best.

- The worse Weight-sensitive distance is the Weight-sensitive Manhattan. The Euclidean and 3-order Minkowski are equal, and the best of them is the Weight-sensitive Exponential-weighted Manhattan, i.e. the *Eixample distance*.

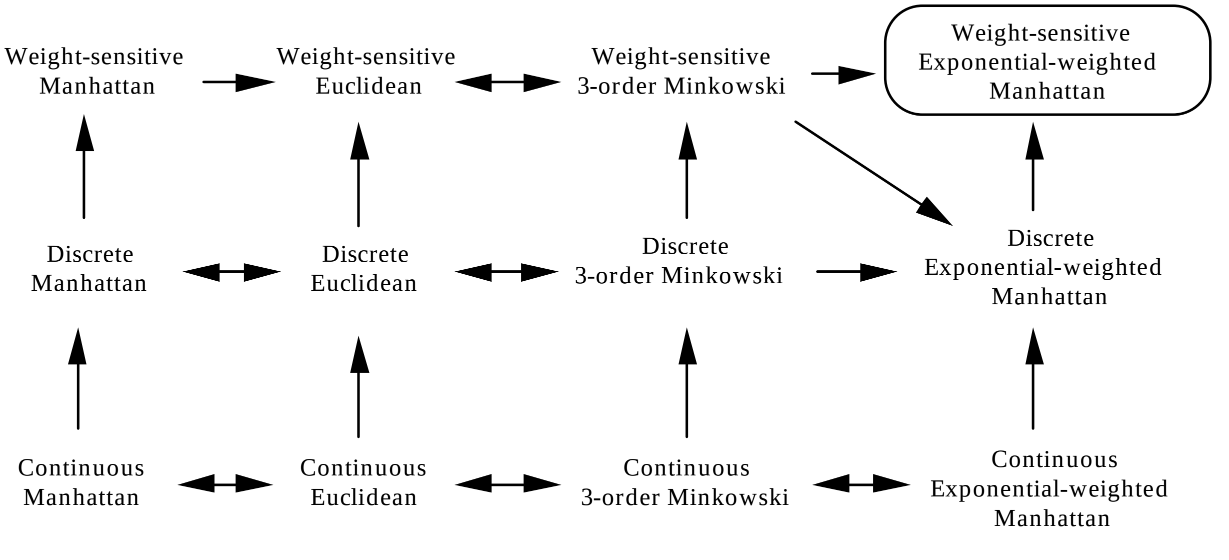


Fig. 8.5. The performance graph of the distances

All theses conclusions are summarized in the figure 8.5, that is a precedence distance graph. In this graph,

A –> B, means that the distance A is "worse" than the distance B.

It shows that the "best" distance from the performed study is the Weight-sensitive Exponential Manhattan distance, i.e. the *Eixample distance*.

8.2.3.2 The CBR competence

In order to test the *competence* of the case-based system, two experiments have been done. In the experiment 1, the case library was seeded with a representative set of initial cases (19) from a previous classification built-up by Linneo$^+$, from real data operation in the WWTP, during the period January 90–September 91. Then, the system was tested against 15 cases formulated by the experts to the system with very good results. In the figure 8.6 it is shown the average marks given by three experts to each one of the 15 cases formulated by them.

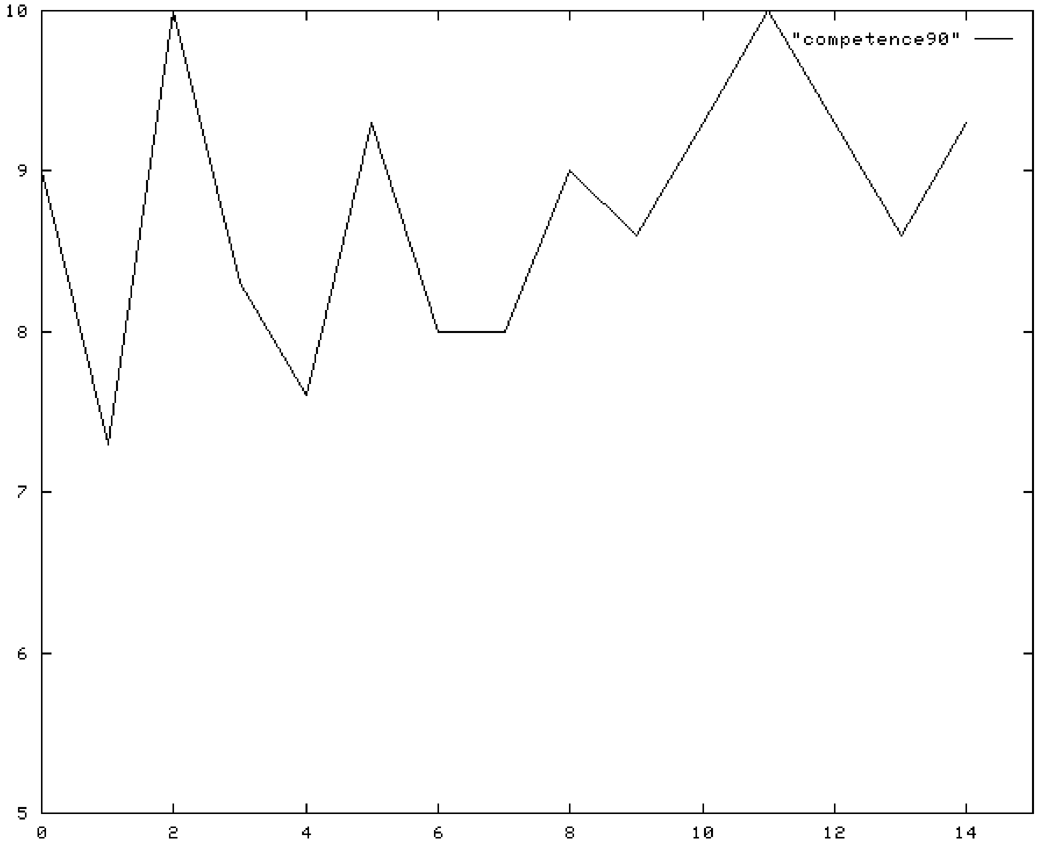


Fig. 8.6. Competence results with the Case Library in the experiment 1

The experiment 2, was to seed the case library with 19 real cases coming from another classification from real data operation in the WWTP, during the period January 94–May 94, and tested against 15 cases formulated by the experts, with also very good results. In the figure 8.7 it is shown the average marks given by three experts to each one of the 15 cases formulated by them.

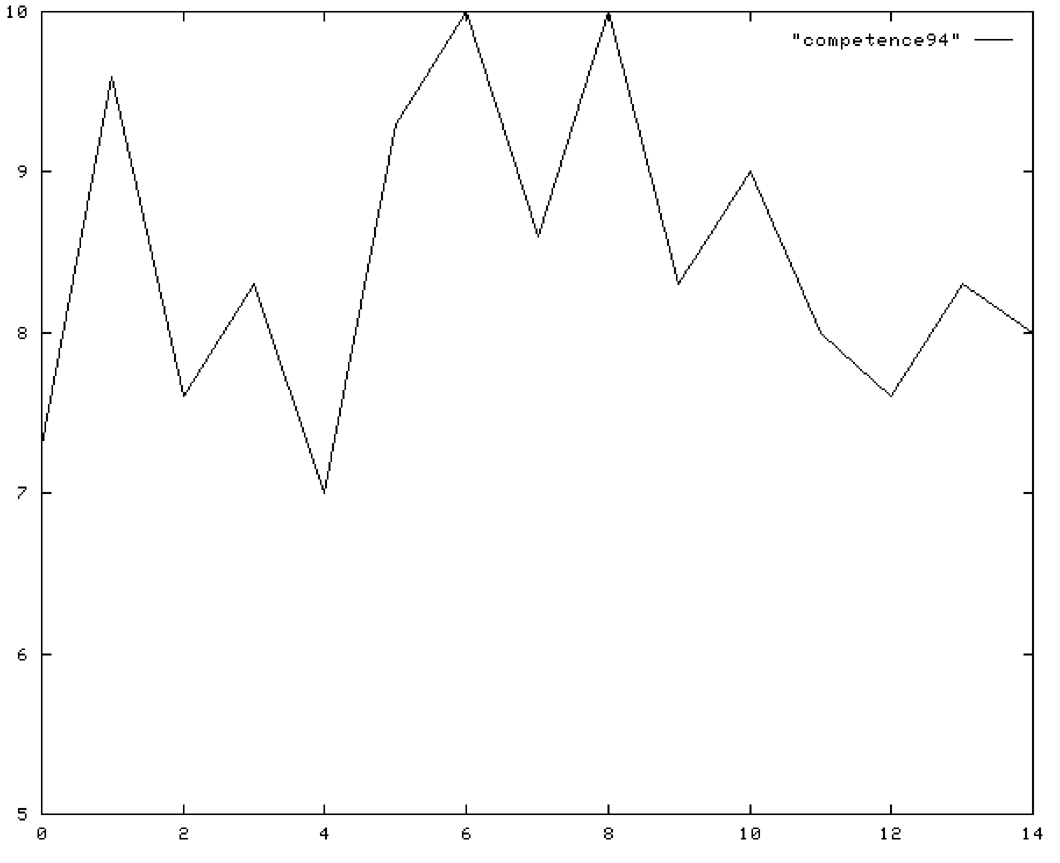


Fig. 8.7. Competence results with the Case Library in the experiment 2

8.2.3.3 The CBR performance

The *performance* of the case-based reasoner, measured both in size of the case library and in response-time was tested with two other experiments. In the third experiment, the same initial case library of the experiment 1 (19 cases from 90/91 real data operation) was tested against a stream of all the 527 real operation cases from the whole period 90/91, with a final size of the case library of 406 cases (77 % of the initial case library). See figure 8.8 for a detailed description.

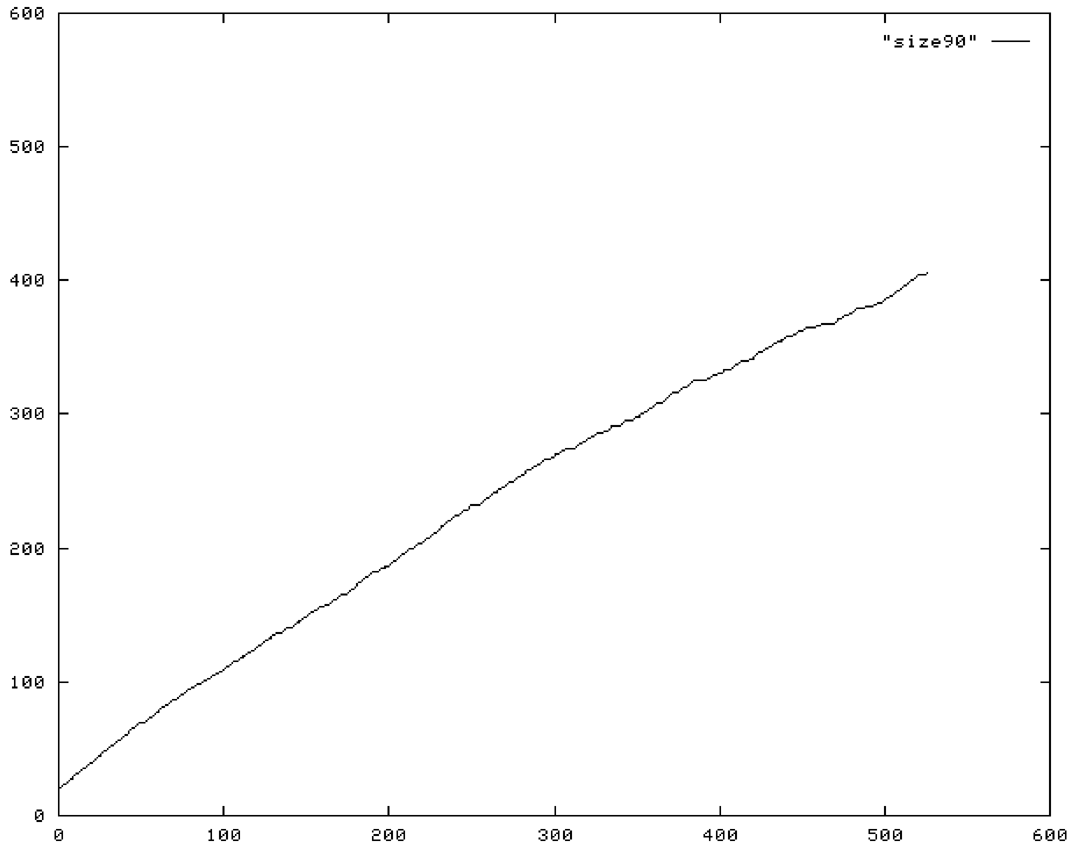


Fig. 8.8. Size evolution of the Case Library in the experiment 1

The percentage of retrieved cases is taken as a measure of the retrieval and matching time used. From the picture of figure 8.9, it is clear that the retrieval time in the case library does not degrade over time, but on the other hand, it maintains or decreases, as a consequence that the retrieval time is almost constant, and the size of the case library is increasing.

In the fourth experiment, the case library was initialized with the same 19 cases from 94 real data operation than in experiment 2, and was tested against a stream of all the 151 real operation cases from the whole period 94, with a final size of the case library of 87 cases (57.6 % of the initial case library). See figure 8.10 for a detailed description.

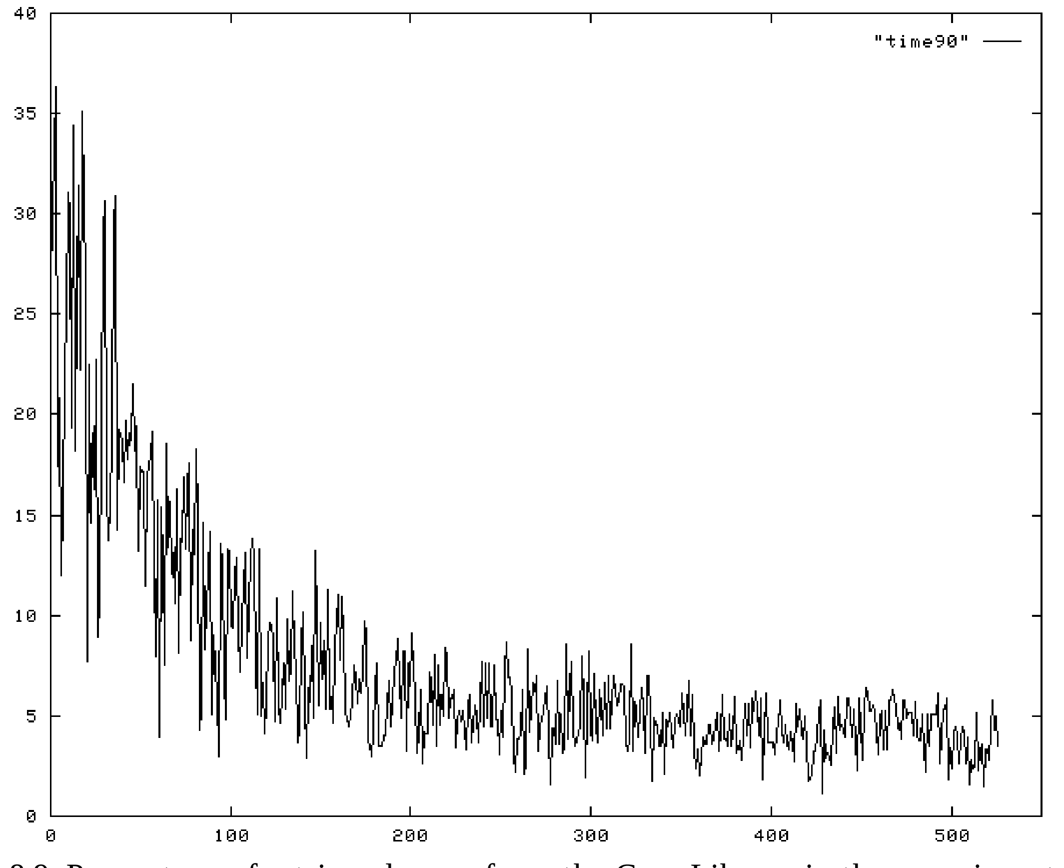


Fig. 8.9. Percentage of retrieved cases from the Case Library in the experiment 1

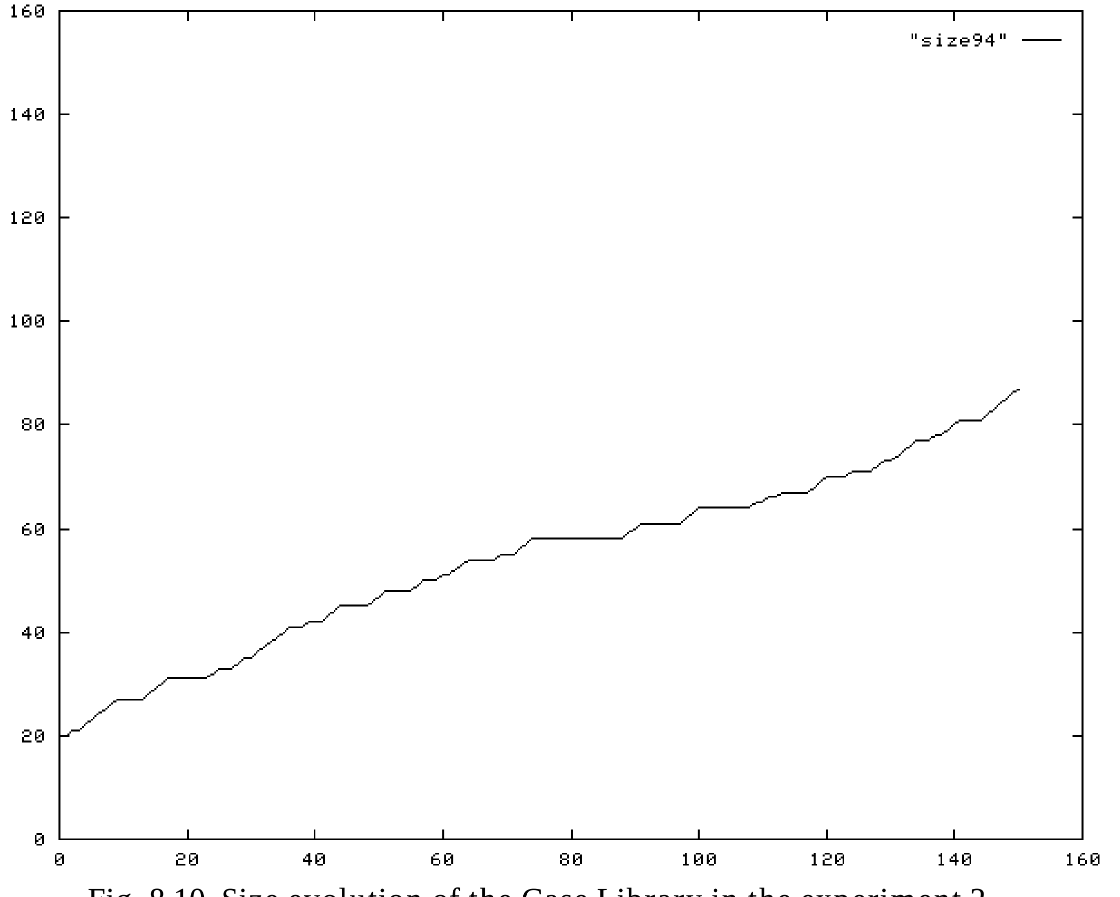


Fig. 8.10. Size evolution of the Case Library in the experiment 2

The percentage of retrieved cases is also taken as a measure of the retrieval and matching time used. From the picture of figure 8.11, it is clear that also in this experiment, the retrieval time in the case library does not degrade over time, but on the other hand, it maintains or decreases, as a consequence that the retrieval time is almost constant, and the size of the case library is increasing.

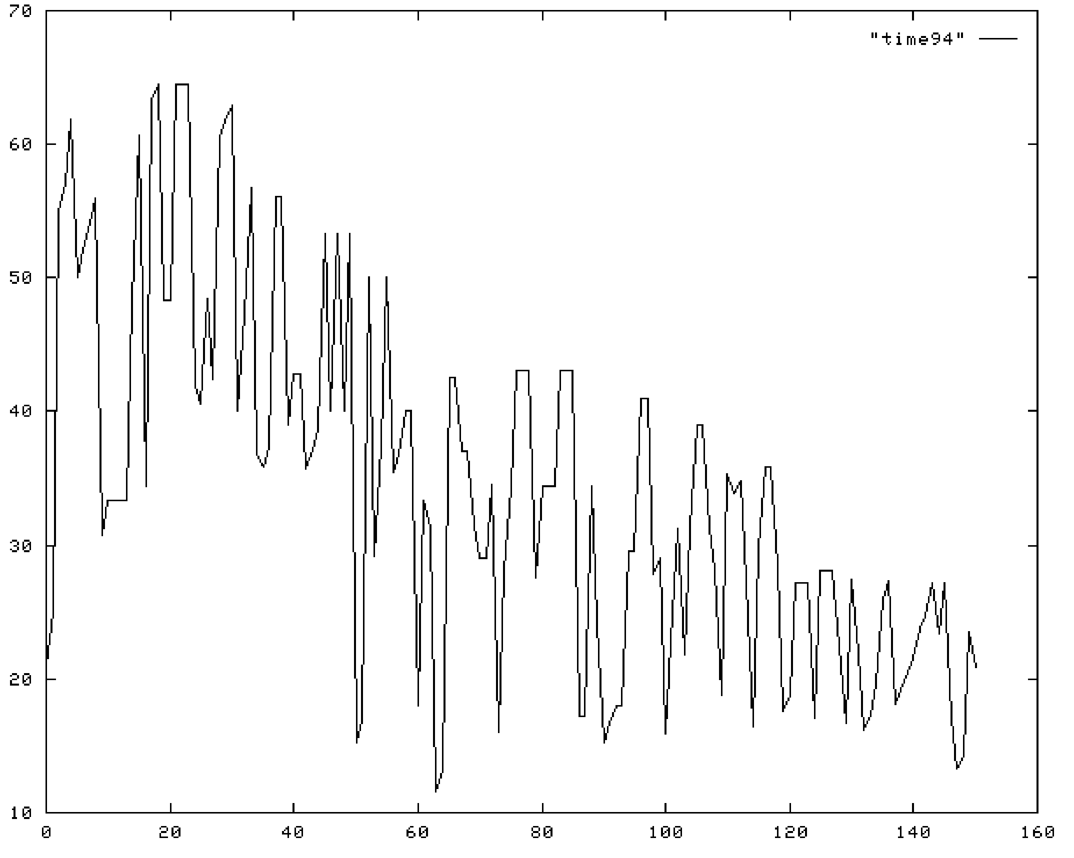


Fig. 8.11. Percentage of retrieved cases from the Case Library in the experiment 2

8.3 Global Evaluation of DAI-DEPUR

In this section, we will describe the global expoerimental evaluation of DAI-DEPUR, at each step of the validation. This evaluation is not still finished, due to the great complexity involved in it.

8.3.1 Simulations

DAI-DEPUR global behaviour was tested in real-time simulation mode against the experts' opinion, yielding very good results. This experimental evaluation was possible because there are several validated process models of the WWTP behaviour that are implemented in the architecture [Serra, 1993].

Currently, we are developing new simulations with the GPS-X simulation package [Hydromantis, 1995]. It is a simulation tool specifically developed for WWTP simulation purposes, and can easily interact with the ACSL simulation language, and with the user through a friendly user interface.

8.3.2 Validation on a Scale Pilot WWTP

A wider evaluation of DAI-DEPUR has been possible due to the construction of a pilot scale WWTP. The chart of the pilot scale WWTP is depicted in figure 8.12. With this pilot scale WWTP, it is possible to create non-standard and/or dangerous situations –those that cannot be tested at the real WWTP lest the environment may be damaged– and to measure our system's performance with real data properly scaled.

Also, we have developed the on-line data acquisition interface for the pilot scale WWTP. See figure 8.13.

On the other hand, there are almost one year of real data from the pilot scale WWTP operation. This feature provides a background to test the competence of DAI-DEPUR. Currently we are testing DAI-DEPUR behaviour with the pilot scale WWTP, with initial good results.

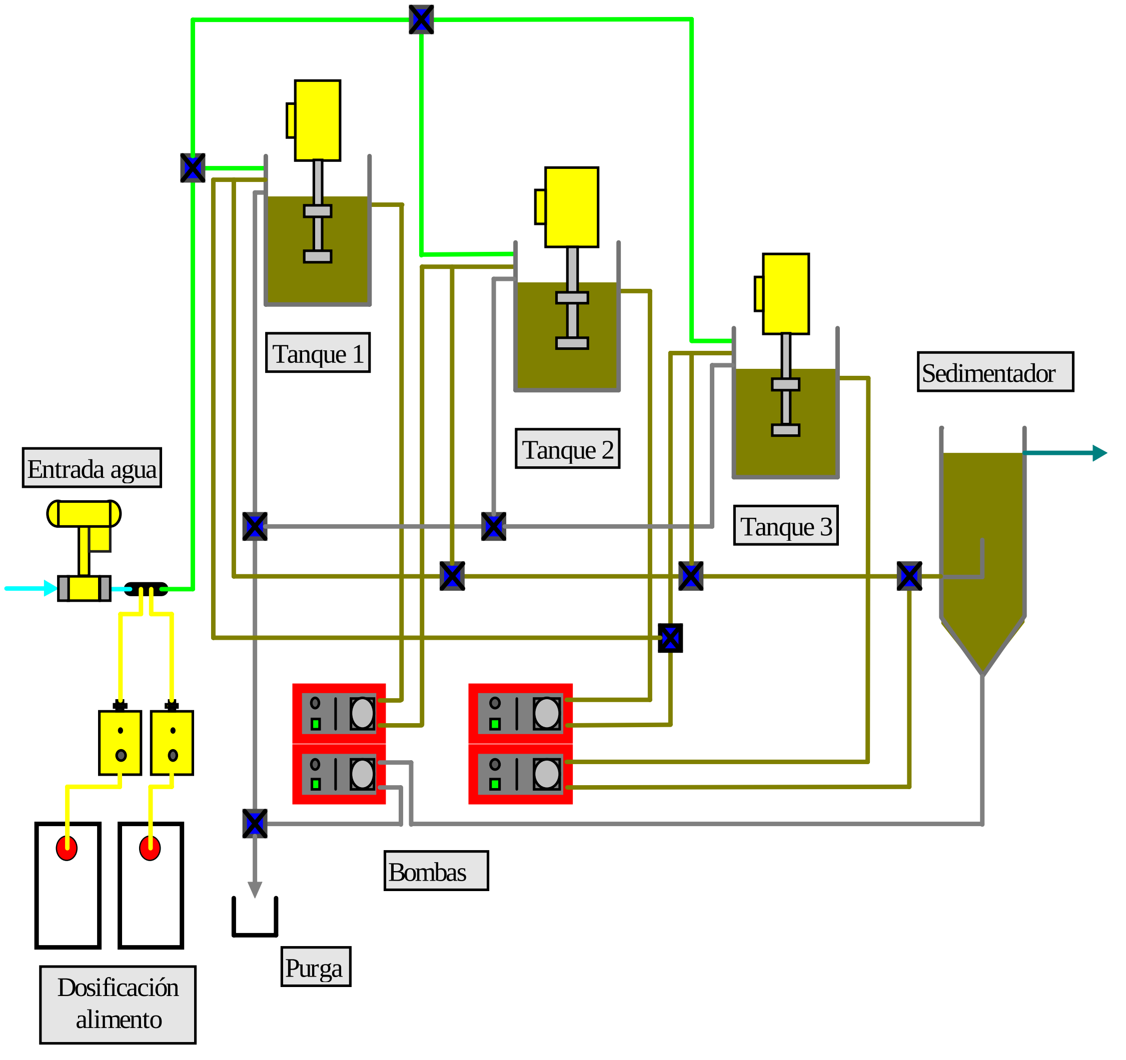


Fig. 8.12. The chart of the pilot scale WWTP

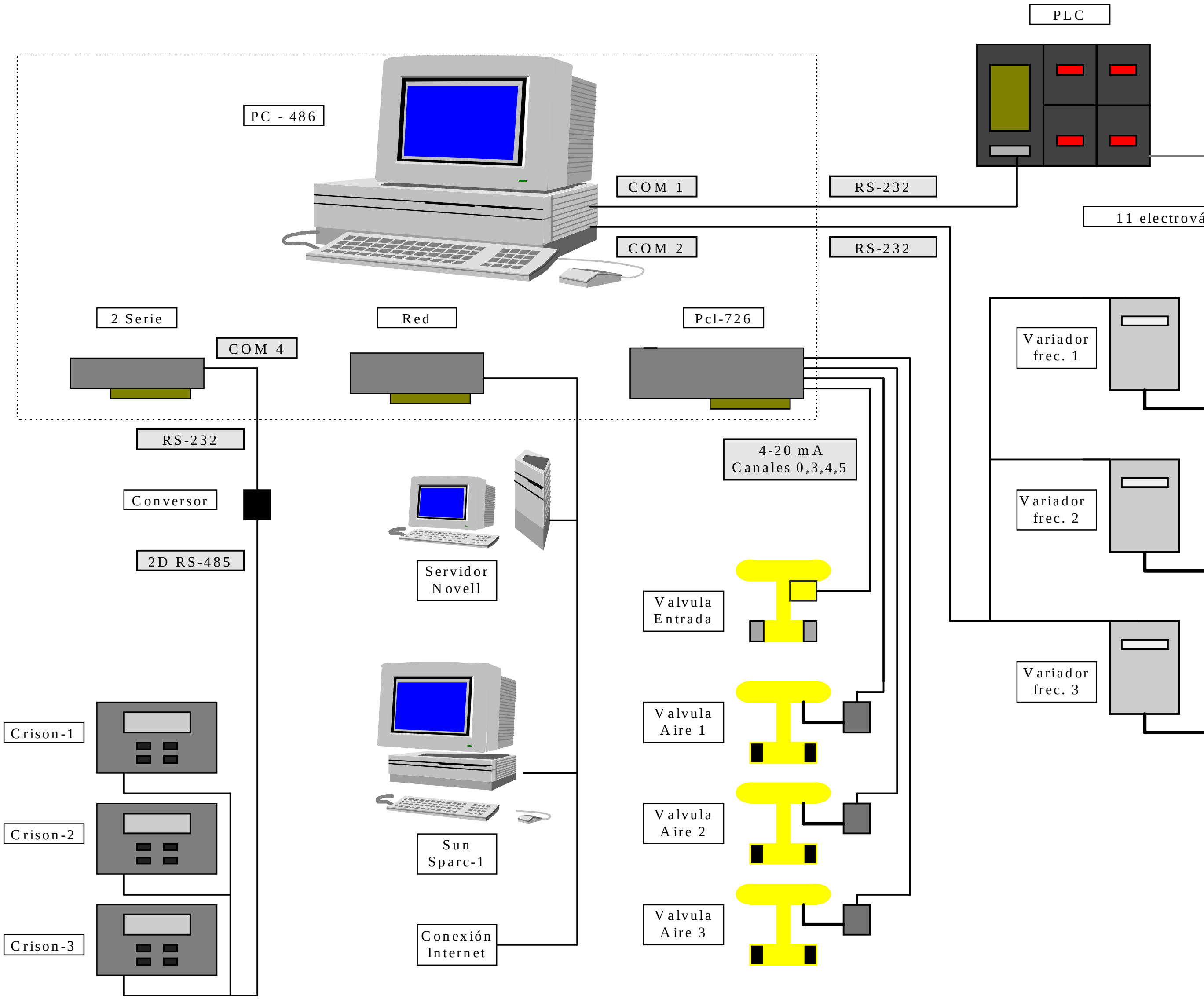


Fig. 8.13. The on-line data acquisition interface of the pilot scale WWTP

8.3.3 Validation on a Real WWTP

We also are working to install DAI-DEPUR in the Cassà de la Selva-LLagostera WWTP. We are developing the on-line data acquisition interface, and adjusting the necessary components of DAI-DEPUR such as gathering a real data stream to derive the initial case library, building the process models of the WWTP, modifying the numerical predictive control algorithm, *etc.*

In a middle-term future (September 1996) it is assumed that this evaluation will be finished. There is an agreement among the "Junta de Sanejament de la Generalitat de Catalunya", as the responsible organism for WWTP management in Catalonia, and our universities to do that work. It will be the ultimate step in the validation of DAI-DEPUR

Chapter 9

Application

9.1 Introduction

The issue of this chapter is to present the main features of the practical use of DAI-DEPUR application, although this is not an exhaustive manual. The next sections describe the execution process of DAI-DEPUR, outlining its main characteristics, and illustrating them by means of some examples. Finally, a brief insight of DAI-DEPUR implementation is detailed.

9.2 Executing DAI-DEPUR

The main outstanding features of DAI-DEPUR execution are the high friendly interaction between the system and the operator that is realized through the user interface, the simulation tool, some inspection facilities, and output displays that make easier the monitoring, control and supervision of a WWTP.

9.2.1 Main menu

To start DAI-DEPUR execution, it is needed to load the G2 shell, and then to select within its main menu the *Start* option (see figure 9.1). The main menu is visualized through a window. There are other options that can be chosen such as *New workspace*, *Get workspace*, *Inspect*, *Load Knowledge Base*, *Merge Knowledge Base*, *Save Knowledge Base*, *System tables*, *Run options*, *Change mode* and *Miscellany*.

Also, some of the options hide new menus with other possible choices such as the *Get workspace*, *System tables*, *Run options* and *Miscellany* commands.

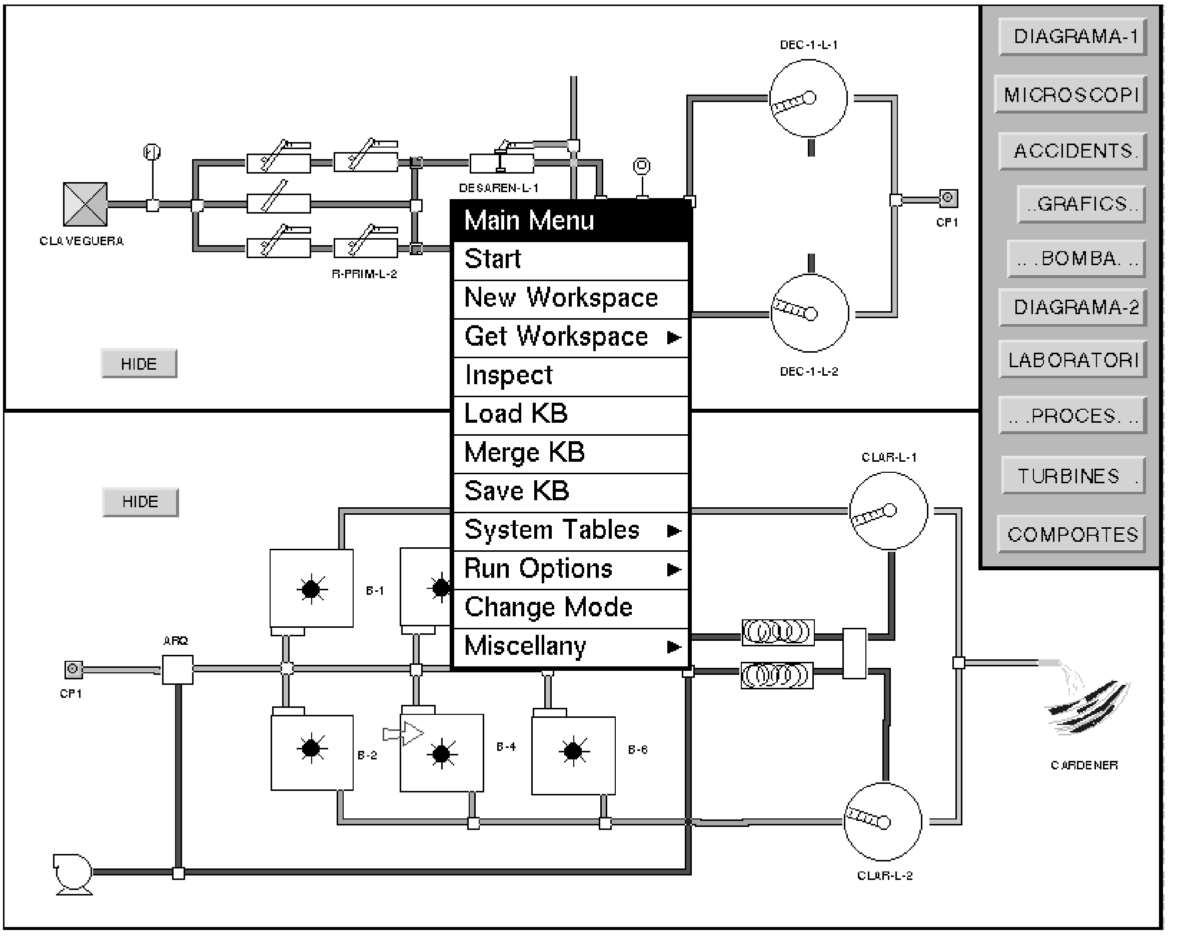


Fig. 9.1. Main menu of G2 shell

When execution starts, all the KBS and other modules and agents are ready to cooperate in the supervision and control of the WWTP.

9.2.2 User interaction

The user interaction is accomplished by means of the *user interface* where there are three main components: a graphical chart of the WWTP that allows to visualize the continuous WWTP operation, a main menu formed of several options, and the displayed information (see figure 9.2).

In the graphical chart of the plant can be followed the evolution of the WWTP under control. Also, the mechanical or electrical faults occurred in the WWTP operation such as turbine aeration motors off, *etc*., are signalled

In the main menu there are several options: to acquire some off-line data coming from analytical tests in the laboratory such as suspended solids (SS), biological oxygen demand (BOD_5), *etc*.; to supply microscopical information from the biomass of the bioreactor; managing and monitoring of some mechanical-electrical equipment such as pumps, aeration turbines, gates, *etc*.; obtaining some graphic information about the evolution of some variables (see 9.2.5).

The displayed information can be some messages generated by the Supervisory-KBS to inform the operator of some discovered alarms, or the results of the diagnosis phase, i.e. the identified global situation, or some output generated by the inspection facility, *etc*.

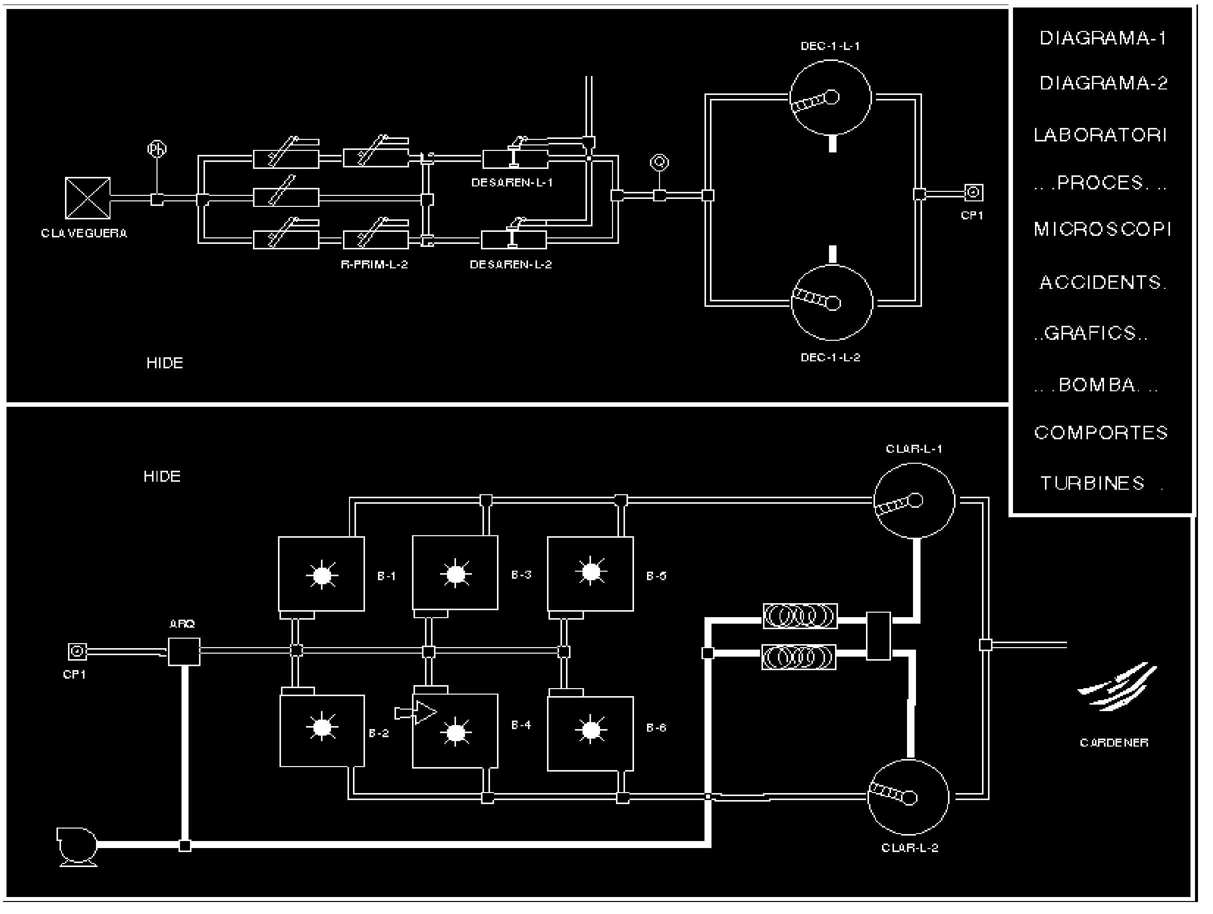


Fig. 9.2. The graphical interface of DAI-DEPUR

9.2.3 Simulation Tool

The simulation tool is provided by the *G2 simulator*. It is a special kind of data server that supplies simulated values for variables and parameters. The simulator uses formulas and procedures to simulate values. It is a safe development and execution tool, which is very important in on-line control tasks. Whenever the simulator is on, it evaluates:

- Every variable that has a specific simulation formula
- Every item for which a generic simulation formula exists
- The main simulation procedure
- All simulation models that are running

To save time the simulator partially evaluates expressions when a Knowledge Base is loaded an when it is edited; this means that it does not need to fully evaluate the expressions while the Knowledge Base is running. It also sorts variables and parameters so that it evaluates the least dependent variables and parameters first, reducing the number of calculations it has to make.

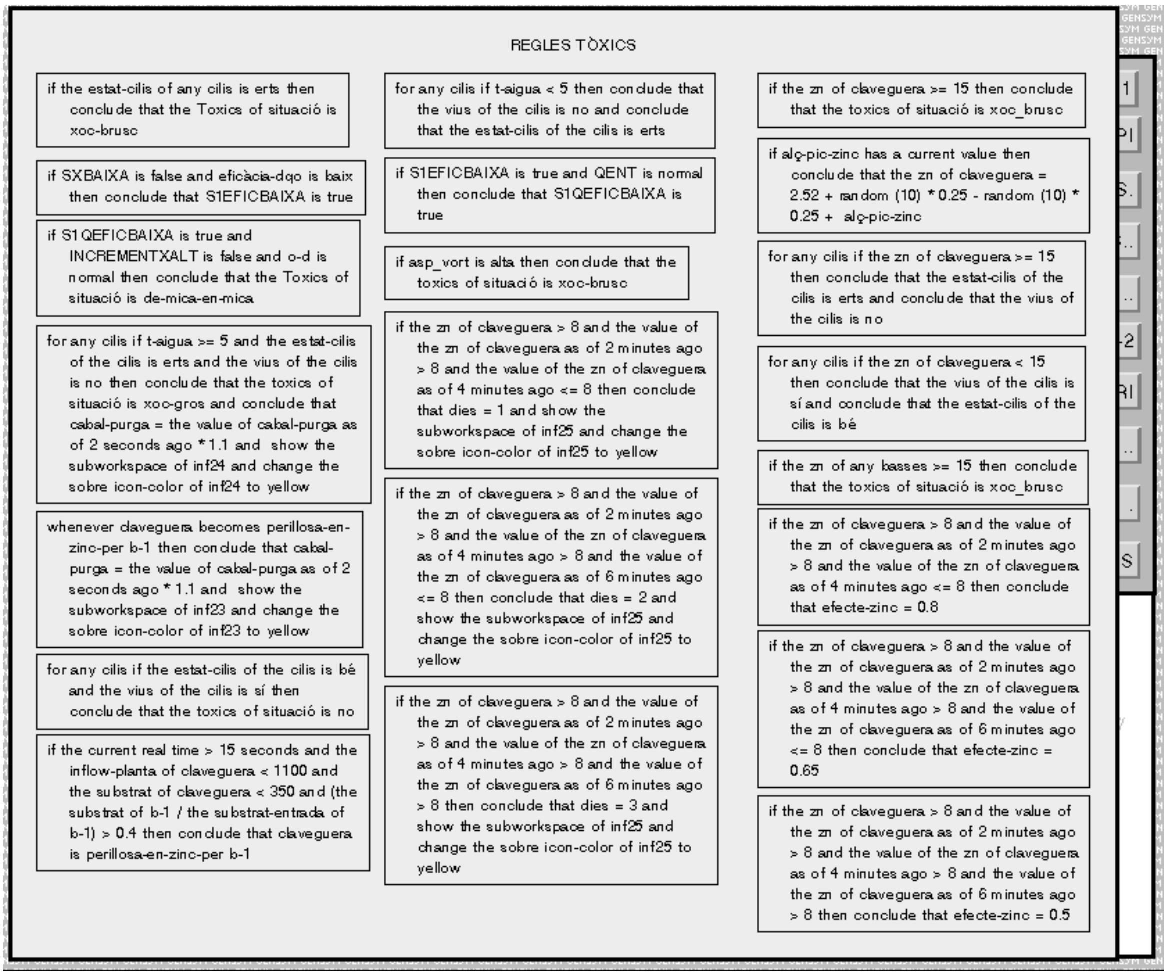
Using the G2 simulator, several components of DAI-DEPUR application can be simulated:

- Test a Knowledge Base
- Run a simulation in parallel with a working process.
- Use simulation models to selectively run and reset parts of a simulation
- Expand DAI-DEPUR safely in the development stage of the application.
- Design control strategies

9.2.4 Inspection Facilities

The *inspection facilities* allow the operator to search trough a Knowledge Base for items based on their type, class, attributes, and location. It can be used off-line, in a development stage of the application to make easy the knowledge engineering task, as well as in execution mode to supply the operator with some information about the rule-based reasoning carried out by the G2 shell. In particular the inspection facility can be used to:

- Search and inspect a particular item of a Knowledge Base.
- Display a table of attribute values.
- Show short representation of items.
- Display a class, module or workspace hierarchies of the different Knowledge Bases. See figures 9.3 (a) and (b).
- Highlight and display any occurrence of an item in a workspace. For example a conclusion, or a premise of an inference rule.
- Display warning messages about errors or conditions that G2 encounters.
- Display trace messages to show the current value of a variable or expression each time it receives one, or when G2 starts and stops evaluating a variable, rule, formula, or function, or when G2 performs each step of evaluating a variable, rule, formula or function.
- Set breakpoints so that G2 can step through an evaluation and halt after each step.
- Highlight the invoked rules, that allows an explanation of the reasoning process carried out in the Knowledge Bases


REGLES TÒXICS
if the estat-cilis of any cilis is erts then conclude that the Toxics of situació is xoc-brusc
if SXBAIXA is false and eficàcia-dqo is baix then conclude that S1EFICBAIXA is true
if S1QEFICBAIXA is true and INCREMENTXALT is false and o-d is normal then conclude that the Toxics of situació is de-mica-en-mica
for any cilis if t-aigua >= 5 and the estat-cilis of the cilis is erts and the vius of the cilis is no then conclude that the toxics of situació is xoc-gros and conclude that cabal-purga = the value of cabal-purga as of 2 seconds ago * 1.1 and show the subworkspace of inf24 and change the sobre icon-color of inf24 to yellow
whenever claveguera becomes perillosa-en-zinc-per b-1 then conclude that cabal-purga = the value of cabal-purga as of 2 seconds ago * 1.1 and show the subworkspace of inf23 and change the sobre icon-color of inf23 to yellow
for any cilis if the estat-cilis of the cilis is bé and the vius of the cilis is sí then conclude that the toxics of situació is no
if the current real time > 15 seconds and the inflow-planta of claveguera < 1100 and the substrat of claveguera < 350 and (the substrat of b-1 / the substrat-entrada of b-1) > 0.4 then conclude that claveguera is perillosa-en-zinc-per b-1
for any cilis if t-aigua < 5 then conclude that the vius of the cilis is no and conclude that the estat-cilis of the cilis is erts
if S1EFICBAIXA is true and QENT is normal then conclude that S1QEFICBAIXA is true
if asp_vort is alta then conclude that the toxics of situació is xoc-brusc
if the zn of claveguera > 8 and the value of the zn of claveguera as of 2 minutes ago > 8 and the value of the zn of claveguera as of 4 minutes ago <= 8 then conclude that dies = 1 and show the subworkspace of inf25 and change the sobre icon-color of inf25 to yellow
if the zn of claveguera > 8 and the value of the zn of claveguera as of 2 minutes ago > 8 and the value of the zn of claveguera as of 4 minutes ago > 8 and the value of the zn of claveguera as of 6 minutes ago <= 8 then conclude that dies = 2 and show the subworkspace of inf25 and change the sobre icon-color of inf25 to yellow
if the zn of claveguera > 8 and the value of the zn of claveguera as of 2 minutes ago > 8 and the value of the zn of claveguera as of 4 minutes ago > 8 and the value of the zn of claveguera as of 6 minutes ago > 8 then conclude that dies = 3 and show the subworkspace of inf25 and change the sobre icon-color of inf25 to yellow
if the zn of claveguera >= 15 then conclude that the toxics of situació is xoc_brusc
if alç-pic-zinc has a current value then conclude that the zn of claveguera = 2.52 + random (10) * 0.25 - random (10) * 0.25 + alç-pic-zinc
for any cilis if the zn of claveguera >= 15 then conclude that the estat-cilis of the cilis is erts and conclude that the vius of the cilis is no
for any cilis if the zn of claveguera < 15 then conclude that the vius of the cilis is sí and conclude that the estat-cilis of the cilis is bé
if the zn of any basses >= 15 then conclude that the toxics of situació is xoc_brusc
if the zn of claveguera > 8 and the value of the zn of claveguera as of 2 minutes ago > 8 and the value of the zn of claveguera as of 4 minutes ago <= 8 then conclude that efecte-zinc = 0.8
if the zn of claveguera > 8 and the value of the zn of claveguera as of 2 minutes ago > 8 and the value of the zn of claveguera as of 4 minutes ago > 8 and the value of the zn of claveguera as of 6 minutes ago <= 8 then conclude that efecte-zinc = 0.65
if the zn of claveguera > 8 and the value of the zn of claveguera as of 2 minutes ago > 8 and the value of the zn of claveguera as of 4 minutes ago > 8 and the value of the zn of claveguera as of 6 minutes ago > 8 then conclude that efecte-zinc = 0.5

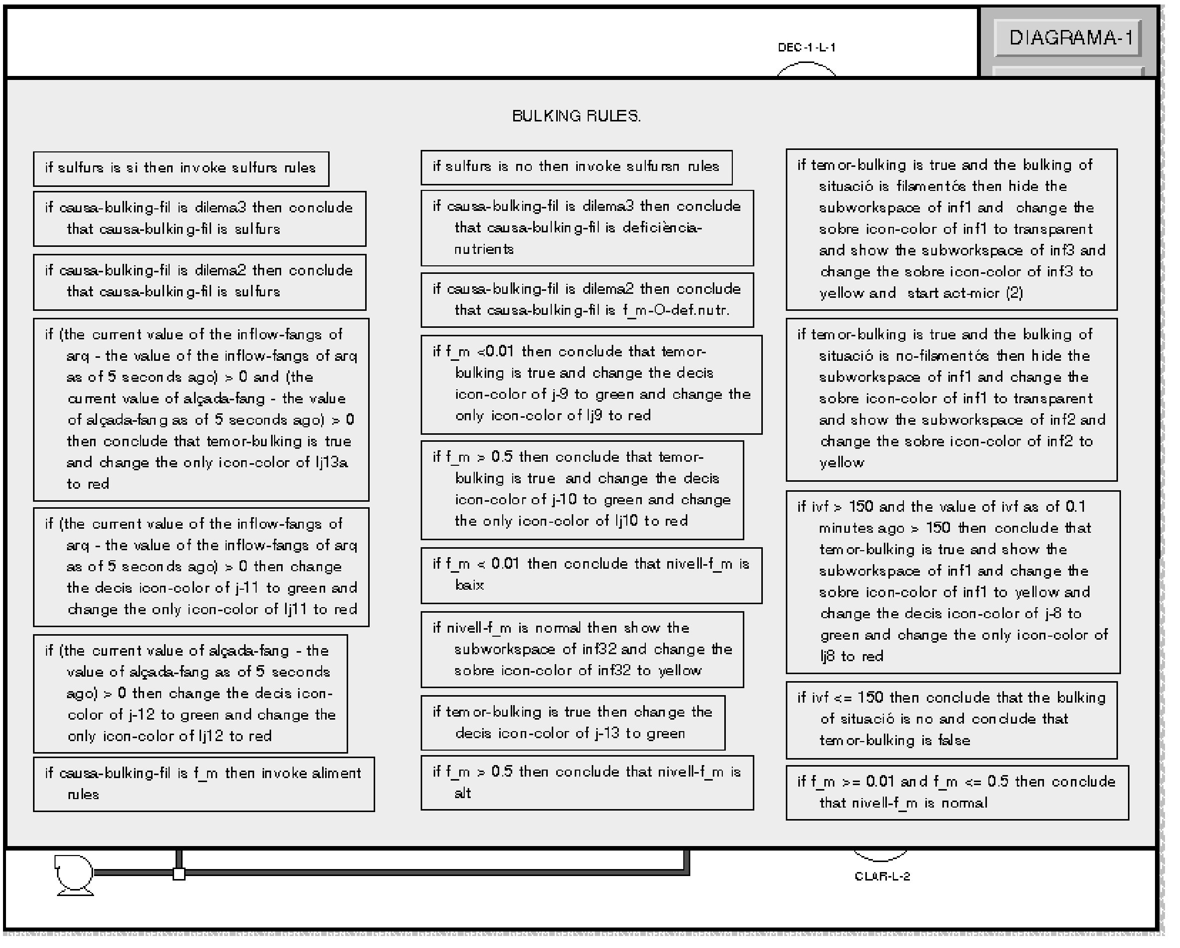


Fig. 9.3 (a) and (b). Output display of inspection results

9.2.5 Output Displays

The *output displays* allow DAI-DEPUR application to visualize some information to the WWTP operator, making the application more user friendly and more reliable for monitoring, control and supervision purposes. With the output displays, the application can show some screens such as:

- Readout-tables, i.e. variables, parameters or expressions, and their values, as well as the scheduler's time.
- Dials, a graphically represented arithmetic values that show the increase or decrease of the numerical magnitudes by means of a circular clock.
- Meters, a graphically represented arithmetic values that show the increase or decrease of the numerical magnitudes by means of a vertical bar.
- Graphs, that plot the histories of one or more variables or parameters. Graphs are ideally suited for showing a history of values (see figure 9.4).

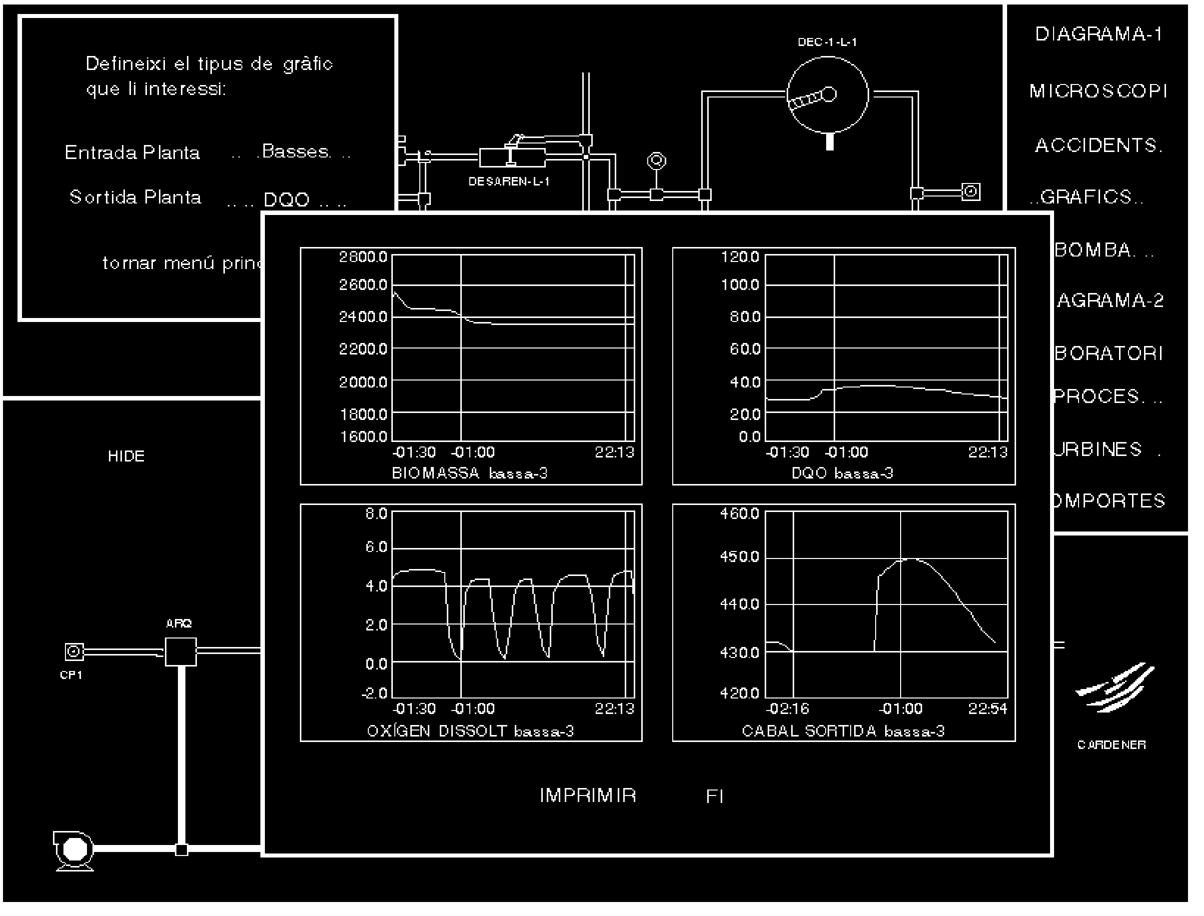


Fig. 9.4. Graphs of some system's variables expressed as a deviation variables

- Charts, that plot any data series against any other data series, or any data series against integers (1, 2, 3, …).
- Freeform-tables, that provide a tabular display of rows and columns. Each cell may display the value of any computed expression.

Also, the results of the inference process carried out by the different Knowledge Bases to diagnose the global current situation of the WWTP, some alarm situations discovered by the Supervisory-KBS, and other useful messages are communicated to the operator by means of some output displays (see figure 9.5).

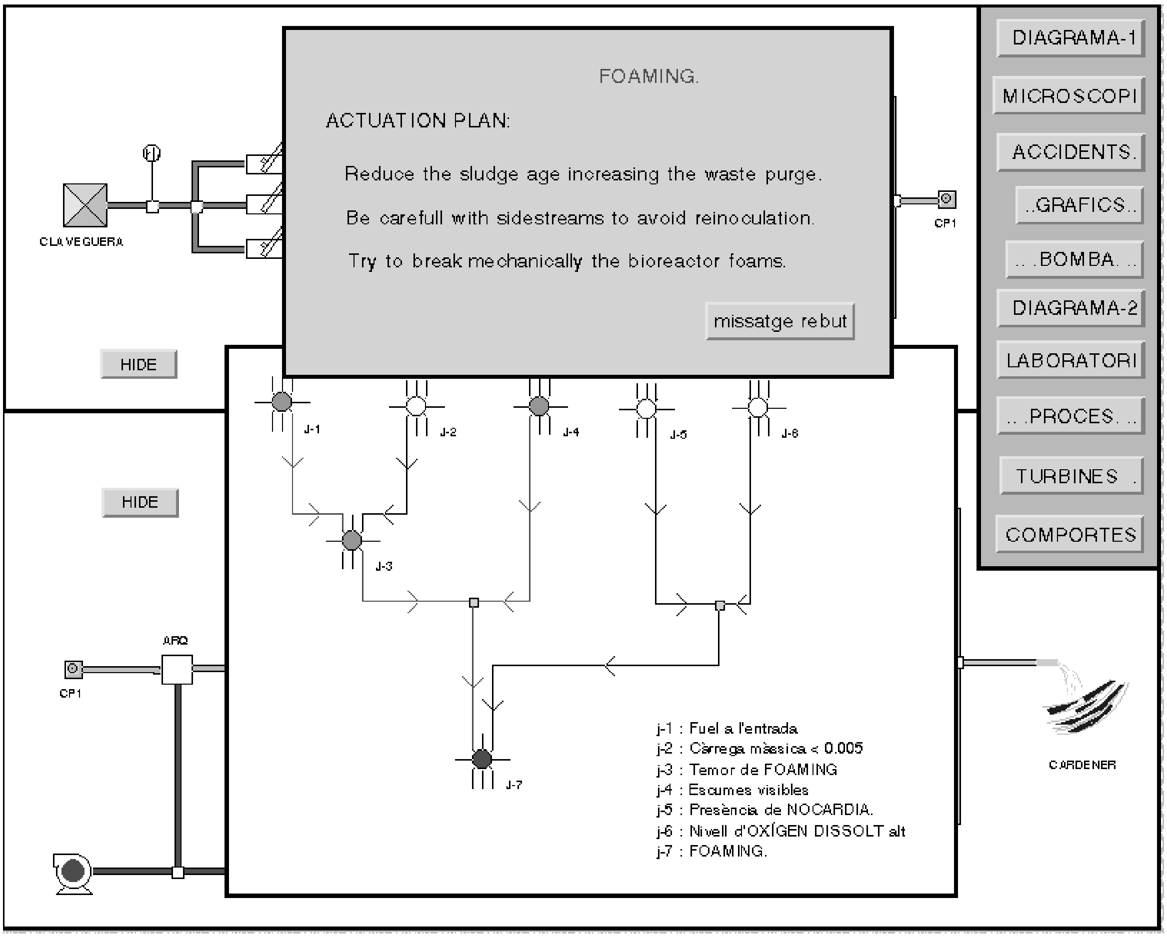


Fig. 9.5. Output display of diagnosis results and actuation

9.3 Examples of Application

Some examples of how works DAI-DEPUR are presented in the next subsections by means of three cases application. Clearly, it can be observed the interaction among the WWTP, the operator, and the different agents and technologies integrated within it.

9.3.1 Uncontrolled Denitrification: *Rising*

In a new supervisory cycle, DAI-DEPUR starts the **local diagnosis task**. The Secondary settler-KBS detects that the Suspended Solids (SS) mixed with the effluent are high, i.e. greater than 35 mg/L allowed by the environmental laws, while the other KBS of the water line does not detect anything anomalous, because the effluent quality is normal, the water temperature is normal (18 ºC), and the DO level in the bioreactor is slightly higher than the prefixed one (2.5 ppm). Another important captured feature is that the sludge age is slowly increasing up to a value of 9 days.

At the same time, the Supervisory-KBS activates the Case-Based Reasoning and Learning (CBRL) agent to retrieve the most similar experienced situation to the current one, and to adapt its actuation plan if it was successful. In the summer of 1991, in the same WWTP, a big inflow of ammonia coming from an industrial waste of a neighbouring industry, combined with old sludge age and a temperature of the mixed liquor (ML) of 20 ºC, provoked the appearance of nitrogen bubbles in the secondary settlers. This phenomenon, known as *Rising*, causes the biomass to lift up to the surface of the settlers. If there is no floating removal equipment, this biomass flows with the treated water providing the effluent with high values of suspended solids (SS) and microorganisms, and decreasing the quantity of microorganisms in the biological process. In the past situation, there was no control action over the WWTP because the ammonia waste and its effects were punctual.

In the **supervision phase**, the Supervisory-KBS perform a deep analysis of all information gathered and inferred by the different local KBS. Combining all local diagnosis, it suspects that the WWTP is in a nitrification process (passing from ammonia to nitrates). This hypothesis is fostered by the high temperature of the water, the old sludge age and the high dissolved oxygen (DO) level in the bioreactor. The obtained nitrates can easily become nitrites in anaerobic conditions, and afterwards become nitrogen gas. This last event would cause the appearance of small bubbles in the secondary settlers, that push the biomass to the surface, preventing it from sedimentation. The Supervisory-KBS to confirm its hypothesis can interact with the operator to require some analytical measures, such as the input and output ammonia values that would detect a high percentage of removal, some observational information such as the presence of bubbles in the secondary settlers, or the V30 analysis in the laboratory.

In this case, both the expert and experiential situation diagnosis agree. In the **validation phase**, the identified situation that would be communicated to the WWTP operator is: *Rising* global situation is very possible. The operator can inspect some of the conclusions reached by the inference engines, which were the relevant data for the diagnosis, which rules have been crucial, the degree of certainty of the reached conclusion, through the explanation module.

The **actuation phase** would consist of the following actuation plan, derived from the expert actuation rules:

1. Increase the recirculation-flow to reduce the quantity of stored sludge in the secondary settlers.
2. Increase the waste-flow in order to diminish the sludge age by means of the washing of the nitrifying microorganisms.
3. Decrease the set point of the dissolved oxygen level in the bioreactor up to 1.5 ppm. This level ensures a sufficient conditions to remove the organic carbon material, but prevent the WWTP from the nitrification of the present ammonia.

Finally, in the **learning task** this new experienced situation and actuation can be stored as a new case into the Case library. In next supervision cycles it must be checked the reliability of the followed actuation plan.

The nitrification-denitrification of ammonia that flows with the water causes the removal of water nutrients. This fact is not only good for avoiding the eutrophyzation of receiving water, but in many cases, also it is needed to fulfil the European environmental legislation. Many activated sludge WWTPs are studying the possibility to include biological denitrification to the conventional process. These WWTPs require a great deal of redesign to cope with the consequences involved in nitrogen source removal [R.-Roda, 1994]. Currently, this uncontrolled nitrification causes them too many problems (*Rising*), and they must prevent from it.

9.3.2 Deficient Sludge Settling: *Filamentous Bulking*

In a new supervisory cycle, within the **local diagnosis task**, the Secondary settler-KBS detects that the sludge level in the secondary settlers progressively increases while the biomass concentration in the recirculation flow decreases, and the sobrenatant is surprisingly clear, due to the formation of a network of filamentous microorganisms that retain small particles (see figure 9.6).

The CBRL agent searches for similar situations in the Case Library. There are several past situations like the current one. All of them are identified as *Bulking situations*, i.e. poor sedimentation of the biomass. But there are several kinds of bulking occurred in the WWTP and it is needed more information to decide to which kind of bulking belongs the current situation.

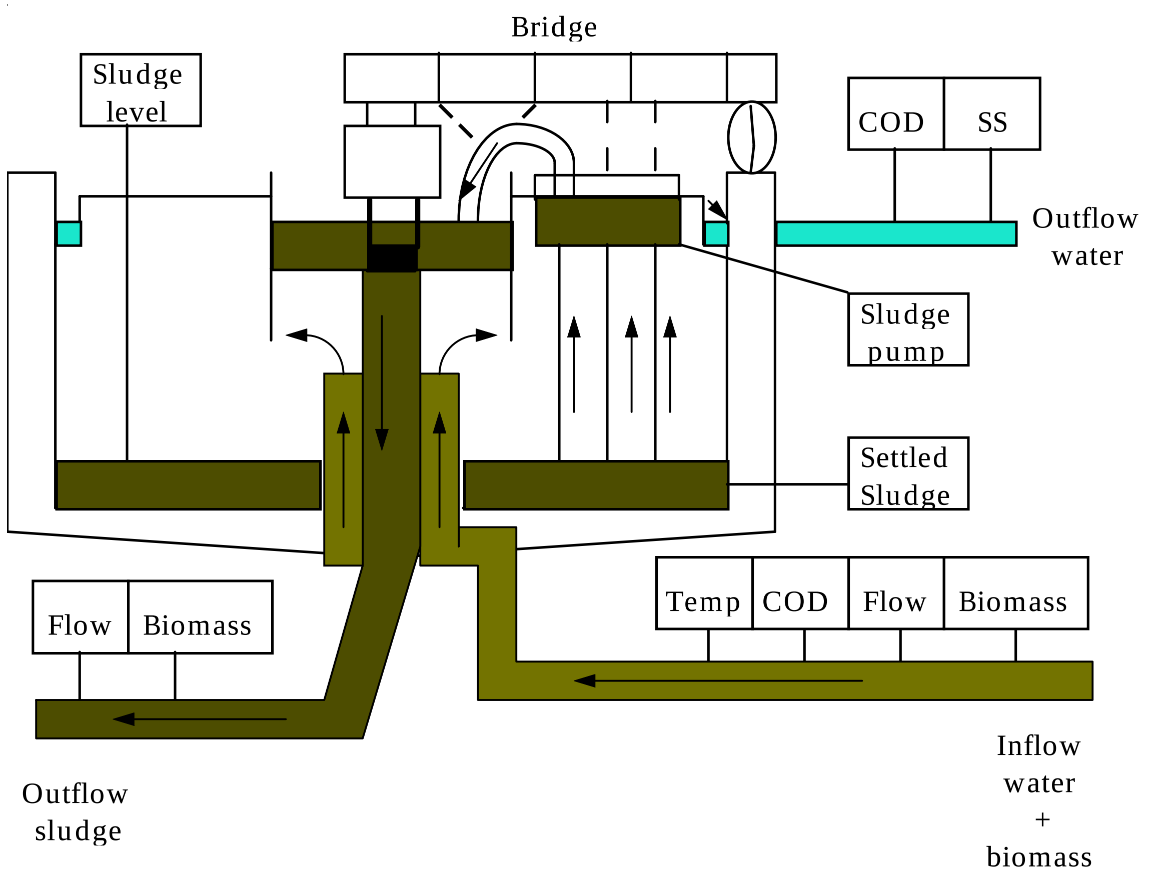


Fig. 9.6. Secondary settler or clarifier

In the **supervision phase**, the Supervisory-KBS perform a deep analysis of all information gathered and inferred by the different local KBS. Combining all local diagnosis, it concludes a *Bulking situation* suspect. Starting the **validation stage**, the Supervisory-KBS requests from the Biological reactor-KBS, the behaviour of the SVI value and microscopical observations to confirm the presence of filamentous microorganisms that causes the *Bulking situation*. Then, it starts a planning strategy to determine the causes for this situation using information coming from several KBS. In particular, the *Bulking* causes considered in the Supervisory-KBS are:

- Low DO level.
- Nutrient deficiency.
- Low F/M ratio.
- Sulphurs presence.
- High pH variability.

CAUSE-IDENTIFICATION RULES

SUPERVISORY-KBS-RULE-012 ::
IF bulking is true AND
S. Natans is in-excess
THEN conclude that
DO-level is low

ACTUATION-INSPECTION RULES

SUPERVISORY-KBS-RULE-051 ::
IF bulking is true AND DO-level is low AND
low-DO-level is not-detected
THEN conclude that
check the status of the DO-sensor

SUPERVISORY-KBS-RULE-052 ::
IF the status of DO-sensor is OK
THEN conclude that
increase the set-point of DO-control

SUPERVISORY-KBS-RULE-053 ::
IF the status of DO-sensor is wrong
THEN
calibrate the DO-sensor AND
keep the DO-control

SUPERVISORY-KBS-RULE-054 ::
IF the status of DO-sensor is not-possible-to-check
THEN
increase the set-point of DO-control AND
keep the alarm-situation AND
send message-34 to operator

Fig. 9.7 Low DO level filamentous bulking rules

As none of the other KBS has signalled a possible *Bulking* cause from those mentioned above, the Supervisory-KBS, it is forced to *believe* that some of the sensors (pH or DO) is sending incorrect values, or the laboratory analysis could be anomalous. To discover the possible cause, the Supervisory-KBS ask the operator for a microscopical observation of the biomass of the bioreactor. Thus, determining which kind of filamentous microorganism is predominant in the sample, will give a quick insight of which can be the cause that has favoured its growing (see 4.1.3.1).

If we consider that the cause of the *Bulking* in the WWTP is low DO level in the bioreactor, then the filamentous microorganism detected would be *S. Natans*, type

1701 or *H.Hydrossis*. Other possibilities would be *Beggiatoa* or *Thiotrix* indicating sulphurs presence; type 021N, 0041, 0675 o *Thiotrix* indicating nutrient deficiency, and *M. Parvicella*, *Nocardia*, *H. Hydrossis*, type 021N, 0041, 0675, 0092, 0581, 0961, 0803 indicating a low F/M ratio.

After the identification of the current situation, the Supervisory-KBS starts the **actuation phase** executing the expert *Bulking* strategy that can be slightly modified by experiential actuation, to avoid biomass decrease in the system. These plans are different for each kind of *Bulking*. In this case would imply an increase of the dissolved oxygen (DO) in the bioreactor, and at the same time, it must inspect why the low DO level has not been already detected, i.e. reflection process. So, it must check DO-sensors, the DO-control (in Numerical Control Knowledge module) or the DO-set-point. The rules involved are shown in figure 9.7.

The main difference between this system and a classical control one is that the second just could act on the set-point of the controller, but DAI-DEPUR may activate or modify the classical control strategy or if necessary looks for operational problems and errors in sensors. In addition, it can retrieve some similar situations that were previously detected, in the past operation of the plant, and update them to help to solve this new situation.

This new experienced situation and actuation brings the possibility to **learn** from it. It can be stored as a new case into the Case library. In next supervision cycles it must be checked the reliability of the followed actuation plan.

9.3.3 *Toxic shock*

The **local diagnosis phase** ends with an alarm discovered by the water line subsystem. It has detected a significative decreasing in the efficiency of the organic material removal. Many causes can be the responsible for this low performance.

The CBRL agent retrieves the most similar past situation. There was a *Solids Shock situation* very similar to the current one, provoked by a heavy metal industrial waste. In the Manresa's WWTP the most common wasted heavy metal is Zinc (Zn).

The Supervisory-KBS disregards several possible causes for the decreasing efficiency of the organic material removal, because the inflow water to the WWTP is normal as well as the organic material concentration in the inflow. Also, the dissolved oxygen in the bioreactor is good and none serious mechanical problem *seems* to affect the WWTP operation. Thus, in this **supervision phase** it is concluded the possibility that a *Solids Shock situation* is occurring in the WWTP. To confirm this hypothesis, the operator must require a microscopical observation of the bioreactor sludge, that would determine the behaviour of different protozoa present in the water. A significative variation in the existing proportion between *aspidisca* and *vorticella* microorganisms would indicate the presence of some toxic substance, because their resistance to the toxic substances is very different. Even, all the protozoa can dead if the shock is very strong and fast.

In this case, also both the expert and experiential situation diagnosis agree. In the **validation phase**, the Supervisory-KBS determines what kind of heavy metal is the responsible of the low depuration efficiency, in order to provide the WWTP with a correct actuation plan. A polarographyc analysis in the WWTP laboratory would allow to find the sending source of the metal. DAI-DEPUR will advise to start an analysis looking for the presence of Zn, because it is the most common heavy metal in their past experiences.

The **actuation phase**, based on an integration of expert and experiential actuation, would increase the waste-flow in 15 % to progressively reduce the concentration of the heavy metal in the WWTP that has been adsorbed by the microorganisms. Another action, that perhaps overcomes the responsibility of the WWTP's manager, would be to locate the origin of the toxic waste, in order to stop it and avoid new toxic substance wastes. Finally, DAI-DEPUR would send a message to the operator, to periodically control the total elimination of the toxic substance in the WWTP.

Finally, the **learning task** can incorporate this new experienced situation and actuation into the Case library. In next supervision cycles it must be checked the reliability of the followed actuation plan.

9.4 Implementation

DAI-DEPUR currently runs on a Sun Sparc workstation. In order to achieve the proposed goals DAI-DEPUR integrates three main tools (see appendix E for a brief description of these tools) to implement the three knowledge/expertise modelling paradigms: ACSL simulation language [Mitchell and Gautier, 1987] used in the predictive dissolved oxygen (DO) control algorithm, the G2 shell [Gensym, 1992; Gensym, 1990] to implement the rule-base reasoning, data base management, temporal reasoning and simulation facilities, and Sun Common Lisp [Sun Microsystems, 1990] to implement the case-based reasoning.

Also, if DAI-DEPUR is controlling on-line a WWTP, it is necessary a supplementary data acquisition computer (PC) connected to the main supervisory computer, to capture the on-line data from the WWTP. However, DAI-DEPUR can be executed in simulation mode, due to the fact that there are some developed models for the WWTP behaviour [Serra, 1993].

The numerical control algorithm uses the ACSL advanced continuous simulation language to predict the dissolved oxygen evolution. This simulation is implemented as a source program in ACSL language that is later compiled to a Fortran program.

The Case-based reasoning approach has been performed by means of some algorithms detailed in chapter 5, by the Case Library structure implemented by a discriminant tree and an ordered discriminant list of the attributes, and by the table of attributes implemented as a hash table. See appendix D for some examples of the Case Library.

Rule-based reasoning and all the other methods and facilities are implemented through the G2 shell. The following elements are integrated in G2:

1) Facts, parameters or variables of the system. It is about 250 elements.
2) Object classes, to define the system. It is about 470 objects.
3) Process modelling described both with formulas, tables, equations, simulation formulas (24 items) and descriptive rules (63 rules)
4) Local diagnosis rules: diagnosis (70 rules), detection of failures (70 rules), and prevention (35 rules).
5) Supervision rules: combination, cause identification and validation rules (70 rules)
6) Control rules: actuation (20 rules) and inspection rules (20 rules).

As the facts, variables or parameters and inference rules have already been explained through this thesis (chapters 4 and 5, 6, 7 respectively), now we only summarize the *object-oriented modelling*.

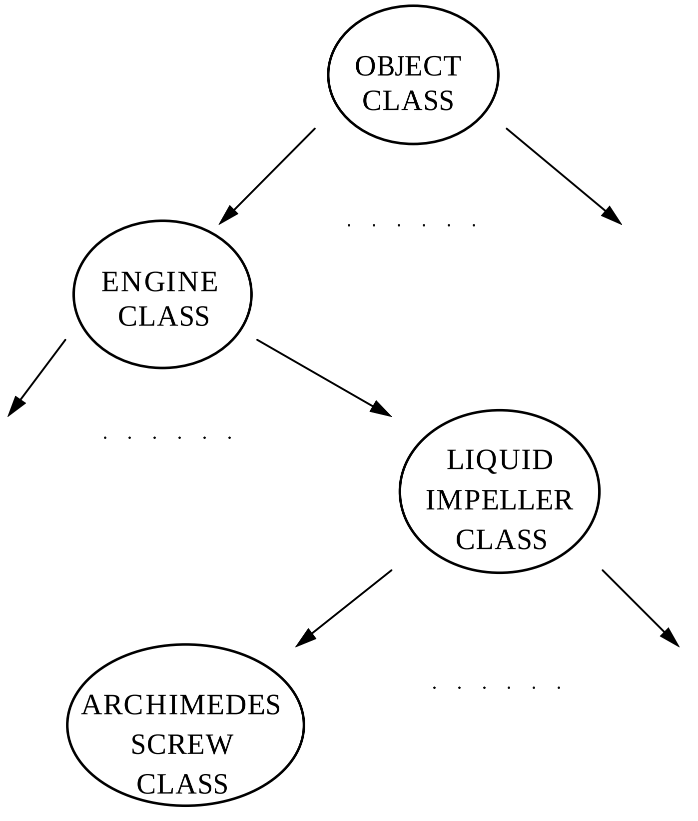


Fig. 9.8 Hierarchy of the object classes

The most common units and objects present in WWTPs have been stored into the object base. Defining a hierarchy of objects and classes of objects saves time and space, as subclasses can inherit attributes from the superior class. For example, in the Manresa's WWTP, there are two types of aeration turbines, Archimedes screws to convey recycled activated sludge and various pumps. All of these types of equipment are used to impel flow, and a superior class of liquid impeller could be defined. Many attributes such as whether the equipment is running or stopped, a maintenance schedule, power consumption, *etc.*, are common to these objects, so they only need to be defined once in the superior class definition.

- Objects
  - Process Equipment
    - Basins
    - Settlers
      - Primary settlers
      - Clarifiers
    - Pools
    - Sources
    - Screens
      - Narrow
      - Wide
      - Manuals
    - Sand removers
    - Manual valves
    - Exterior
    - Thickeners
    - Flotation units
    - Digesters
    - Tanks
    - Belt filter presses
  - Motors
    - Turbines
      - 1 speed
      - 2 speed
    - Arquimedes screw
    - Pumps
    - Electrovalves
  - Biomass
    - Fungi
    - Microorganisms
    - Protozoa
      - Flagellates
      - Free-swimming ciliates
      - Ciliates
        - Vorticella
        - Aspidisca
      - Rotifers
  - Floc
  - Microorganism-characteristics
  - Situation
  - Color-button

Table 9.1. Object base for the Manresa's WWTP

As it is shown in figure 9.8, the liquid impeller class would have a superior class, for example, the engine class, which would have a superior object class, that could be the highest level defined. These means that a few powerful generic rules can be written which apply to a wide range of objects avoiding duplication. Besides, new attributes can be added to any class if more aspects about the class are considered or deleted otherwise. An example of a definition is:

```
(AERATION-BASINS
    (Attributes
        INFLOW
        OUTFLOW
        BIOMASS
        SUBSTRATE (COD)
```

DISSOLVED OXYGEN
BIOMASS-AT-THE-INPUT
SUBSTRATE-AT-THE-INPUT
DO-AT-THE-INPUT
OXYGEN-UPTAKE-RATE (OUR)
pH
GATE-OPEN-OR-CLOSE
FULL-EMPTY))

In table 9.1, the classes of objects considered in DAI-DEPUR are shown. Once the classes are defined different instances of them are created and connected between them to build the scheme of the WWTP.

Chapter 10

Conclusions and Future Work

If one cannot explain what he has been doing, then his work is not useful.
Erwin Schrödinger

10.1 Research discussion

An *integrated* and *distributed multi-level supervisory* architecture, DAI-DEPUR, for wastewater treatment plants supervision in real-time has been designed and implemented. It was designed to overcome the insufficiency of classical Chemical Engineering control methods, and some shortcomings of Knowledge-Based Systems, specially when coping with complex real-world problems. After the evaluation stage of DAI-DEPUR, we can claim that the architecture fulfils the specified requirements.

This *integrated* architecture approach has several advantages that make it more powerful than other classical technologies applied to wastewater treatment plants, as well as to other complex domains:

- If hard real-time deadlines are needed, they could be satisfied through *concurrent computation* of all agents. In the supervision of a WWTP, as explained in 2.1.2.1, there are not hard real-time deadlines, but hard output quality constraints. So, concurrent computation is not a requirement for WWTP supervision.

• *Modularity* and *extendibility*. The several KBS and other distributed problem solving processes that cooperate in the WWTP supervision, provide DAI-DEPUR with a good modularity and extendibility, that make easier the maintenance, spread and debugging of the system.

• *Learning from experience*. DAI-DEPUR can learn from past experienced situations in a certain WWTP, by means of storing these past experiences in its memory. This experiential knowledge allows DAI-DEPUR to increase its competence and to prevent itself from making the same mistakes than in the past, by means of a case-based reasoner.

• *Reusability*. DAI-DEPUR can be exported to any wastewater treatment plant with similar technology, with minor changes. This reusability aim was already outlined by the experts in the first stage of the research, as a relevant feature to be accomplished.

• *Knowledge acquisition module* to solve the bottleneck of the knowledge acquisition. With the semi-automated knowledge acquisition tool, LINNEO$^+$, DAI-DEPUR captures the expert knowledge provided by the experts and the literature.

• *Integration* of problem solving capabilities, reasoning, learning, on-line data acquisition, numerical control, simulation tasks, *etc.*, in a single system.

• DAI-DEPUR provides a *multi-disciplinary* integrated way to supervise a WWTP. The experience from different people and different backgrounds such as Microbiologists, Chemical Engineers, Expert operators, Control Engineers and Computer scientists has been used.

• Microbiological, qualitative and quantitative information available from the WWTP operation is taken into account to supervise the process.

• The WWTP can be controlled in *normal situations* (mathematical control) , in *usual abnormal situations* (expert control), and in *unusual abnormal situations* (experiential control).

On the other hand, the design and implementation of the architecture is more difficult and complex than other single-technology approaches applied to WWTP supervision and control.

10.2 Contributions

Mainly, the power of DAI-DEPUR relies on the *integration* of several techniques at the knowledge / expertise level, on the *multi-level* structure for the different kinds of knowledge and reasoning, and on the *distributed problem solving* scheme.

From what has been previously described, one can state that the combination of the three paradigms at the *knowledge/expertise level*, allows the system to model the numerical control knowledge (supplied by a predictive control algorithm), to model the subjective knowledge (supplied by the experts) as well as the objective knowledge (supplied by the real operation of the concrete plant under control)[1]. This integration presents some advantages that are the addition of the own ones from rule-based reasoning, from case-based reasoning, from dynamic learning, and from semi-automatic knowledge acquisition, that are the methods acting at the knowledge / expertise level:

- DAI-DEPUR supports reasoning *in a poor understood and ill-structured domain*, where other kinds of reasoning like model-based reasoning or algorithmic reasoning could not be possible or easily formulated.

- DAI-DEPUR is able to learn from previously solved problems and to adapt the available experiential knowledge over the domain (*dynamic learning environment*).

- DAI-DEPUR overcomes the *brittleness* of KBS' in coping with unforeseen situations (not previously considered by the expert general knowledge), trying to solve them by means of the most closely situation in the Case library.

- DAI-DEPUR captures the knowledge provided by the experts (*knowledge acquisition*) which is very important –although subjective– to get a central corpus of knowledge about the domain.

---

[1] So that it is commonly known –and has been showed out in the performed study [Sànchez *et al.*, 1995e]– that not all the considered situations by the experts occur in the practice, and vice versa, certain situations not taken into account by the experts can occur in the WWTP.

• DAI-DEPUR deals either with prototypical situations (*general knowledge*) or with idiosyncratic or exceptional ones (*specific knowledge*).

• Due to the dynamic learning environment, the system is able to adapt itself to a specific wastewater treatment plant. Thus, DAI-DEPUR is portable to another plant if we supply the system with a case library formed of a set of specific cases (operating situations of the concrete WWTP), which can be obtained semi-automatically from real operational data.

The multiple components of DAI-DEPUR are more powerful than systems using a single technology applied to wastewater treatment plants as knowledge-based approaches [Lapointe *et al.*, 1989; Maeda, 1989], statistical process control techniques [Novotny *et al.*, 1990], fuzzy controller methods [Alex *et al.*, 1994; Czoagala and Rawlik, 1989], *etc.*

On the other hand, our research has contributed to the emerging research activity that joins Artificial Intelligence with Environmental Science to preserve our planet. In recent past years, we have knowledge of the two first Workshops until now, merging these two fields:

• The AAAI'94 Workshop on Environmental Applications of AI. Seattle, USA.
• The IJCAI'95 Workshop on Artificial Intelligence and the Environment. August, Montréal. Canada, where we participated [Sànchez *et al.*, 1995b]

10.3 Future work

There are some future work lines to be considered. Some of these features have appeared as the consequence of DAI-DEPUR evaluation. All of these future lines are intended to improve the competence, usefulness and the performance of DAI-DEPUR:

• Integration of a new agent to support a new feature to be considered in the cleansing process of a WWTP: the *nitrification problem*.

• Other new research direction points to consider that Knowledge Bases would not be static but dynamic ones. Inference rules could be adapted as well as CBRL agent update the Case Base accordingly with new experience.

• The research in WWTP field has to be in mind the developing of an *automatic pattern recognition* of microbiological images to capture this useful information to the system [Dellepiane *et al.*, 1992]. To this end, we are studying the integration of DAI-DEPUR with VEX-93 [Valdés *et al.*, 1994] to include on-line identification of microorganisms to automatically capture this qualitative information into the supervisory system.

• As it was explained in 5.4.1.2, it can be interesting to study the feasibility of establishing a *situation transition network*, that could capture the temporal transitions from one situation to another, in order to prevent the WWTP operation from an upset situation. It could establish a model-based reasoning to model the causal relationships among WWTP situations.

• As shown in 8.2.3, the size of the case library may become too large. As we mentioned in 5.4.2.3, we are developing a new strategy to add the new cases to memory, based on a *relevance measure* of that new cases to be added. This measure states that a case is relevant if the minimum distance to all the cases stored at the corresponding leaf-node is greater or equal to γ.

• We are defining, within the introspective reasoning of the system, the notion of exceptional and redundant cases, in order to establish a reliable deletion policy to *forgetting* cases.

• Another feature we are studying to improve in the CBRL agent is having different discriminating attributes for different cases. The idea is that in the retrieval phase, first, the case-based system would search within a previously established classification to identify which kind of case it is coping with. For each established class (*meta-case*) there would be a set of specific discriminating attributes and a different case library. Thus, there will be a partition of the cases by means of its similarity to previous established *meta-cases*.

• The application of the data set and knowledge bases as retrofitting information to optimization in the design of wastewater treatment plants [Bañares-Alcántara and Ponton, 1992] could be another useful line of future work.

• In a middle-term future, we think that we have to by-pass the G2 shell, and implement ourselves all the facilities and techniques that it supports for two reasons. First for a practical and economical fact: if DAI-DEPUR has to be used in real WWTPs, the economic cost of the G2 shell, is too much expensive for the responsible institutions of WWTP management. Secondly, because the integration of the different processes would be less difficult than now.

• Other challenge would be to test DAI-DEPUR architecture in a different real-time process control within a complex real-world domain.

Appendix A

Glossary

This glossary provides the reader with accurately defined concepts and main used abbreviations related to Wastewater Treatment Plants and Chemical Engineering terminology:

Acidity. The capacity of a solution to react with hydroxyl ions. Acidity is measured by titration with a standard alkaline solution (base) to a specified end point. Typically, it is measured in milligrams of calcium carbonate per litre.

Activated sludge. Sludge withdrawn from a secondary clarifier following the activated sludge process. Activated sludge consists mostly of biomass, with some inorganic settleable solids. Return sludge is recycled to the head of the process; waste (excess) sludge is removed for conditioning.

Activated sludge loading. The kilograms (pounds) of biochemical oxygen demand (BOD) in the applied liquid per unit volume of aeration capacity or per kilogram (pound) of activated sludge per day.

Activated sludge process. A biological wastewater treatment process by which a mixture of wastewater and activated sludge is agitated and aerated. The activated sludge is subsequently separated from the treated wastewater (mixed liquor) by sedimentation and wasted or returned to the process as needed.

Advanced waste treatment. Any physical, chemical, or biological treatment process used to accomplish a degree of treatment greater than that achieved by secondary treatment (see also tertiary treatment).

Aeration. The initiation of contact between air and liquid by **one** or more of the following methods: (a) spraying the liquid in the air; (b) bubbling air through the liquid; (c) agitating the liquid to promote surface absorption of air.

Aeration period. The time, usually expressed in hours, during which mixed liquor is subjected to aeration in an aeration tank while undergoing activated sludge treatment.

Aeration tank. A tank in which wastewater or other liquids are aerated (also called aeration basin).

Aerator. A device that brings air and liquid into intimate contact (see also aeration).

Aerobes. Organisms that live only in aerobic conditions.

Aerobic. Living or occurring in an environment containing oxygen (such as an aeration tank).

Aerobic respiration. The breakdown or organic substances by aerobes in the presence of oxygen.

Air lift. A device for raising liquid by injecting air in and near the bottom of a riser pipe submerged in the liquid to be raised.

Air-lift pump. A pump used for lifting activated sludge from the aeration basin or clarifier to waste or return activated sludge. Fine pressured air bubbles are discharged to the water at the bottom of the basin or clarifier. The bubbles reduce the density of the water at the bottom, and the denser surrounding water pushes up in the discharge pipe to the outlet (also called air-lift or air-lift returns).

Algae. Photosynthetic, microscopic plants that can seriously deplete oxygen in the presence of sunlight.

Alkalinity. The capacity of a solution imparted by carbonates; bicarbonates; hydroxides; and occasionally borates, silicates, and phosphates to neutralize acids. Alkalinity is measured in milligrams of equivalent calcium carbonate per litre.

Ammonia. A chemical combination of hydrogen (H) and nitrogen (N) occurring extensively in nature and expressed as NH_3.

Ammonia-nitrogen. The quantity of elemental nitrogen present in the form of ammonia (NH_3).

Amoeba. Small, one-celled organism using pseudopodia (false feet) for movement (see Sarcodina).

Amperometric titration. The electronic detection of the equivalence point in a titration, through observation of the change in diffusion current at a suitable applied voltage as a function of the volume of titrating solution.

Anaerobes. Organisms that live in the absence of oxygen.

Anaerobic. A condition in which no oxygen is-available in the environment (for example, a septic clarifier).

Anaerobic respiration. The breakdown of organic substances in the absence of oxygen.

Bacteria. A group of universally distributed, rigid, essentially unicellular microscopic organisms lacking chlorophyll. Bacteria perform a variety of biological treatment processes, including biological oxidation, nitrification, and denitrification.

Bacterial examination. The examination of wastewater to determine the presence, number, and identity of bacteria. Also called bacterial analysis.

Biodegradable. The destruction of organic materials by organisms and wastewater treatment systems.

Biomass. The amount (usually measured in kilograms or pounds) of biological material contained in the treatment system.

Biochemical oxygen demand (BOD). (1) The quantity of oxygen used in the biochemical oxidation of organic matter in a specified time, at a specified temperature, and under specified conditions. (2) A standard test used in assessing wastewater strength.

Biochemical oxygen demand (BOD) load. The BOD content, usually expressed in kilograms (pounds) per unit of time, of wastewater passing into a waste treatment system or to a body of water.

Centrifuge. Mechanical device used to separate solids from water using a centrifugal force (commonly called spin test when used as a process control test).

Chemical oxygen demand (COD). A quantitative measure of the amount of oxygen required for the chemical oxidation of carbonaceous (organic) material in wastewater, using inorganic dichromate or permanganate salts as oxidants in a 2-hour test.

Ciliated protozoa. Small, one-celled organisms possessing cilia (hairlike projections used for movement).

Clarification. Any process or combination of processes, the primary purpose of which is to reduce the concentration of suspended matter in a liquid. The term was formerly used as a synonym for settling or sedimentation. In recent years, the latter terms are preferable when describing the settling process.

Clarified wastewater. Wastewater from which most of the settleable solids have been removed by sedimentation (also called settled wastewater).

Complete-mix. Activated sludge process whereby wastewater is rapidly and evenly distributed throughout the aeration tank, unlike the conventional aeration process (plug flow).

Concentration. (1) The amount of a given substance dissolved in a unit volume of solution or applied to a unit weight of solids. (2) The process of increasing the suspended solids per unit volume of sludge as by sedimentation.

Contact stabilization. A modification of the activated sludge process using a short contact time for adsorption of BOD followed by a long contact time for synthesis or stabilization by bacteria.

Contact time. The period of time a substance remains in a basin or tank (see detention time).

Conventional aeration. Process design configuration whereby the organic loading in the aeration tank is higher at the influent end than at the effluent end. The flow passes through a serpentine system of tanks, typically side-by-side, before passing on to the clarifier (also called plug flow).

Core sampler. A long, slender pole with a foot valve at the bottom end that allows the depth of the sludge blanket to be measured (also called sludge judge).

Declining growth phase. Period of time between the log-growth phase and endogenous phase, where the amount of food is in short supply, leading to ever-slowing bacterial growth rates.

Denitrification. The anaerobic biological reduction of nitrate nitrogen to nitrogen gas. Also, removal of total nitrogen from a system (see also nitrification).

Depth of blanket (DOB). The level of sludge, typically measured in metres (feet), in the bottom of the clarifier (see also sludge blanket).

Design flow. Engineering guidelines that typically specify the amount of influent flow that can be expected on a daily basis over the course of a year. Other design flows can be set for monthly and peak flows.

Detention time. The period of time a wastewater flow is retained in a basin or tank for storage or completion of physical, chemical, or biological reaction (see also contact time).

Dissolved oxygen (DO). The oxygen dissolved in wastewater, usually expressed in milligrams per litre, or percent of saturation.

Dissolved solids. Solids in solution that cannot be removed by filtration; for example, NaCl and other salts that must be determined by evaporation (see also total dissolved solids).

Dynamic equilibrium. See population dynamics.

Effluent. Wastewater partially or completely treated, flowing out of a basin, treatment plant, or industrial treatment plant.

Effluent quality. The physical, biological, and chemical characteristics of wastewater or other liquid flowing out of a basin, pipe, or treatment plant.

Effluent standard. Specification of the allowable concentration or mass of a constituent that may be discharged.

Effluent stream. A stream of treated wastewater.

Endogenous phase. See endogenous respiration.

Endogenous respiration. The internal digestion of stored food within the organism occurring when the external food sources are limited.

Excess sludge. The sludge produced in an activated sludge treatment process, or any other process that requires sludge recirculation, that is not needed to maintain the process and is withdrawn from circulation (also called waste sludge or waste activated sludge [WAS]).

Extended aeration. A modification of the activated sludge process that provides for aerobic sludge digestion within the aeration system. The process includes the stabilization of organic matter under aerobic conditions. Effluent contains finely divided suspended matter and soluble matter.

Extended aeration process. A modification of the activated sludge process using long aeration periods to promote aerobic digestion of the biological mass by endogenous respiration.

Facultative. The ability of an organism to live in aerobic or anaerobic conditions.

Filamentous growth. Intertwined, threadlike biological growths, characteristic of some species of bacteria, fungi, and algae. Such growths reduce sludge settleability and dewaterability.

Filamentous organisms. Bacterial, fungal, and algal species that grow in thread-like colonies, resulting in a biological mass that will not settle and may interfere with drainage through a filter.

Filamentous sludge. Activated sludge characterized by excessive growth of filamentous bacteria, resulting in poor sludge settling.

Floc. Collections of smaller particles agglomerated into larger, more easily settleable particles through chemical, physical, or biological treatment (see also flocculation).

Flocculation. In water and wastewater treatment, the agglomeration of colloidal and finely divided suspended matter after coagulation by gentle mechanical or hydraulic stirring. In biological wastewater treatment where coagulation is not used, agglomeration may be accomplished biologically.

Flow. The movement of water or other fluids from place to place.

Flow rate (Q). Volume of liquid that passes through a cross-section of conduit in a given time; measured in such units as kilograms per hour, cubic metres per second, litres per day, or gallons per day.

Flow recording. Documentation of the quantity of rate of the flow.

Food-to-microorganism ratio (F:M). In the activated sludge process, the loading rate expressed as kilograms (pounds) of BOD_5 per kilogram (pound) of mixed liquor or mixed liquor volatile suspended solids per day.

Free swimming ciliates. Mobile, one-celled organisms using cilia (hairlike projections) for movement.

Fungi. Small non-chlorophyll-bearing plants lacking roots, stems, or leaves. Fungi occur in, among other places, water, wastewater, or wastewater effluents and grow best in the absence of light.

High-purity oxygen. A modification of the activated sludge process using relatively pure oxygen and covered aeration tanks in a conventional flow arrangement.

High-rate aeration. A modification of the activated sludge process whereby the mixed liquor suspended solids loadings are kept high, allowing high food-to-microorganism (F:M) ratios and shorter detention times.

Hydraulic loading. The amount of wastewater applied to a given treatment process, usually expressed as volume per unit time, or volume per unit time per unit surface area.

Influent. Wastewater flowing into a basin, treatment plant, or treatment process (see antonym effluent).

Inorganic compounds. All of those combinations of elements that do not include organic carbon.

Inorganic matter. Mineral-type compounds that are generally nonvolatile, noncombustible, and nonbiodegradable. Most inorganic type compounds, or reactions, are ionic in nature; therefore, rapid reactions are characteristic.

Kjeldahl nitrogen test. A standard analytical method used to determine the concentration of organically bound ammonia nitrogen state.

Log growth phase. The initial stage of bacterial growth, during which there is a plentiful supply of food, causing bacteria to grow at the maximum rate.

Maximum flow. The greatest volume of influent to a treatment plant within a given time period (see peak flow).

Mean cell residence time (MCRT). Average time a given unit of cell mass stays in the activated sludge aeration tank. Mean cell residence time is typically calculated as the ratio of the total mixed liquor suspended solids in the aeration tank to that of wastage.

Mean flow. The arithmetic average of the discharge at a given point or station on the line of flow for some specified period of time (see design flow).

Mechanical aeration. (1) The mixing, by mechanical means, of wastewater and activated sludge in the aeration tank of the activated sludge process to bring fresh surfaces of liquid into contact with the atmosphere. (2) The introduction of atmospheric oxygen into a liquid by the mechanical action of a paddle, paddle wheel, spray, or turbine mechanism.

Mechanical aerator. A mechanical device used for introducing atmospheric oxygen into a liquid (see also mechanical aeration).

Metazoa. Group of animals having bodies composed of cells differentiated into tissues and organs and usually a digestive cavity lined with specialized cells.

Microbial activity. The activities of microorganisms resulting in chemical or physical changes.

Microbiology. The study of microscopic organisms of living matter and their processes.

Microorganisms. Microscopic organisms, either plant or animal, that are invisible or barely visible to the naked eye. Examples are algae, bacteria, fungi / protozoa, and viruses.

Microscopic examination. (1) The examination of wastewater to determine the presence and amount of plant and animal life such as bacteria, algae, and protozoa. (2) The examination of wastewater to determine the presence of microscopic solids. (3) The examination of microbiota in process water, such as the mixed liquor in an activated sludge plant.

Minimum flow. (1) The flow occurring in a stream during the driest period of the year (also called low flow). (2) The lowest quantity of influent to a treatment plant or within a sewer within a give time period (see antonym peak flow).

Mixed liquor. A mixture of raw or settled wastewater and the activated sludge process.

Mixed liquor suspended solids (MLSS). The concentration of suspended solids in activated sludge mixed liquor, expressed in milligrams per litre.

Mixed liquor volatile suspended solids (MLVSS). That fraction of the suspended solids in activated sludge mixed liquor that can be driven off by combustion at 550°C (1022°F); indicates the concentration of active microorganisms available for biological oxidation.

Moving average. A tool used in trend analysis for determining patterns or changes in treatment processes. For example, a 7-day moving average would be the sum of the datum points for 7 days divided by 7.

Nematodes. Any of a phylum (Nematoda) of elongated cylindrical worms parasitic in animals or plants or free-living in soil or water.

Nitrate. An oxygenated form of nitrogen, typically written (NO_3^-) (see nitrogen).

Nitrification. The oxidation of ammonia nitrogen to nitrate nitrogen in wastewater by biological or chemical reactions (see also denitrification) .

Nitrite. An intermediate oxygenated form of nitrogen typically written (NO_2^-) (see nitrogen).

Nitrogen. An essential nutrient often present in wastewater as ammonia, nitrate, nitrite, and organic nitrogen. The concentrations of each form and the sum, total nitrogen, are expressed as milligrams per litre elemental nitrogen. Nitrogen is also present in some groundwater as nitrate and in some polluted groundwater in other forms.

National Pollutant Discharge Elimination System (NPDES) permit. Permit that is the basis for the monthly monitoring reports required by most states in the U.S.A.

Organic. Volatile, combustible, and sometimes biodegradable chemical compounds containing carbon atoms (carbonaceous) bonded together and with other elements. The principal groups of organic substances found in wastewater are proteins, carbohydrates, and fats and oils (see antonyms inorganic compounds and inorganic matter).

Organic loading. The amount of organic material, typically measured as BOD_5, applied to a given treatment process; expressed as weight per unit time per unit surface area or unit weight.

Organic matter. Chemical substances of animal or vegetable origin or, more correctly, containing carbon and hydrogen.

Overflow rate. The settling velocity of particles removed in an ideal basin if they enter at the surface; one of the criteria for the design of settling tanks in treatment plants. Overflow rate is expressed as volume of flow per unit water surface area of the tank (see also surface overflow rate).

Oxidation ditch. A secondary wastewater treatment facility that uses an oval channel with a rotor placed across it to provide aeration and circulation. The screened wastewater in the ditch is aerated by the rotor and circulated at about 0.3 to 0.6 m / s (1 to 2 ft / sec) (see also secondary treatment).

Oxygen demand. The quantity of oxygen used in the oxidation of substances in a specified time, at a specified temperature, and under specified conditions.

Oxygen uptake rate (OUR). The oxygen used during biochemical oxidation, typically expressed as milligrams O_2 per litre per hour in the activated sludge process.

Peak flow. The maximum rate of influent flow a treatment plant expects to receive during a specified time period (for example, peak hourly, peak daily, peak monthly).

pH. A measure of the hydrogen-ion concentration in a solution. On the pH scale (0 to 14), a value of 7 at 25°C (77°F) represents a neutral condition. Decreasing values, below 7, indicate an increasing hydrogen-ion concentration (acidity); increasing values, above 7, indicate a decreasing hydrogen-ion concentration (alkalinity).

Pin floc. Small floc particles that settle poorly.

Plug flow. See conventional aeration.

Population dynamics. The everchanging numbers of microscopic organisms within the activated sludge process (also called dynamic equilibrium).

Positive displacement pump. A type of pump in which the water is induced to flow from the source of supply through an inlet pipe and inlet valve. Water is brought into the pump chamber by a vacuum created by the withdrawal of a piston or piston-like device which, on its return, displaces a certain volume of the water contained in the chamber and forces it to flow through the discharge valves and discharge pipes.

Primary effluent. The liquid portion of wastewater leaving primary treatment.

Primary sludge. Sludge obtained from a primary settling tank.

Primary treatment. (1) The first major treatment in a wastewater treatment facility, usually sedimentation but not biological oxidation. (2) The removal of a substantial amount of suspended matter but little or no colloidal and dissolved matter. (3) Wastewater treatment processes usually consisting of clarification with or without chemical treatment to accomplish solids-liquid separation.

Protozoa. Small animals, including amoebae, ciliates, and flagellants.

Publicly owned treatment works (POTW). In general, another name for wastewater treatment plants.

Raw influent. Wastewater before it receives any treatment.

Receiving water. A river, lake, ocean, or other watercourse to which wastewater or treated effluent is discharged.

Respiration. The intake of oxygen and discharge of carbon dioxide during the process of bacterial decomposition of organic materials.

Respiration rate. See specific oxygen uptake rate (SOUR).

Return sludge. Biomass produced in the activated sludge process that is recycled to the head of the process to promote more complete biological oxidation (also called return activated sludge [RAS]).

Rotifers. Minute, multicelled aquatic animals possessing a circular set or sets of ciliate resembling wheels.

Sarcodina. Species of amoebae found in wastewater.

Secchi disk. A visual inspection tool to measure the clarity or turbidity of the effluent.

Secondary effluent. (1) The liquid portion of wastewater leaving secondary treatment. (2) An effluent that contains not more than 30 mg/L each of BOD_5 and suspended solids.

Secondary treatment. (1) Typically, a level of treatment that produces removal efficiencies of 85% for biochemical oxygen demand (BOD) and suspended solids. (2) Sometimes used interchangeably with the concept of biological wastewater treatment, particularly the activated sludge process. This term is commonly applied to treatment that consists chiefly of clarification followed by a biological process, with separate sludge collection and handling.

Secondary wastewater treatment. Wastewater treatment processes usually consisting of primary treatment and biological oxidation using activated sludge or trickling filtration, followed by clarification. Secondary wastewater treatment is typically interpreted as the attainment of a least 85% removal. or secondary effluent concentrations less than 30 mg/L, of both BOD_5 and suspended solids on a monthly average basis.

Septic. See anaerobic.

Settleability test. Determination of the settleability of solids in a suspension by measuring the volume of solids settled out of a measured sample over a specified interval of time; typically reported in millilitres per litre (see settleometer).

Settleometer. A 2-litre or larger beaker used to conduct the settleability test.

Sludge. (1) The accumulated solids separated from liquids such as during wastewater processing. (2) The removed material resulting from flocculation, sedimentation, and/or biological oxidation of wastewater (see also activated sludge).

Sludge age. The average residence time of suspended solids in a biological waste treatment system, equal to the total weight of suspended solids in the system divided by the total weight of suspended solids leaving the system per unit of time (typically per day).

Sludge blanket. Accumulation of sludge hydrodynamically suspended within an enclosed body of wastewater (see depth of blanket).

Sludge judge. See core sampler.

Sludge solids. Dissolved and suspended solids in sludge

Sludge volume index (SVI). The ratio of the volume in millilitres (cubic inches) of sludge settled from a 1000-mL (60-cu in.) sample in 30 minutes to the concentration of mixed liquor in milligrams per litre multiplied by 1000.

Solids. In wastewater treatment, any dissolved, suspended, or volatile substance contained in or removed from wastewater.

Solids inventory. The amount of sludge in the treatment system typically expressed in kilograms (tons). The inventory of plant solids can be tracked through use of a mass balance set of calculations.

Solids loading. The amount of solids applied to a treatment process per unit time per unit volume.

Solids retention time (SRT). The average time of retention of suspended solids in a biological waste treatment system, equal to the total weight of suspended solids leaving the system per unit of time (typically per day).

Specific oxygen uptake rate (SOUR). Measures the microbial activity in the biological system. It is typically expressed as milligrams O_2 per hour per gram of volatile suspended solids (VSS) (also called respiration rate).

Spin test. See centrifuge.

Stabilization. In waste treatment, a process used to equalize wastewater flow composition before regulated discharge.

Stalked ciliates. Small, one-celled organisms possessing cilia (hairlike projections used for feeding) but that are not mobile.

Step feed aeration. See step feed.

Step feed. A procedure for adding increments of settled wastewater along the line of flow in the aeration tanks of an activated sludge plant (also called step feed aeration).

Straggler floc. Large (6-mm [0.25-in.] or larger) floc particles that have poor settling characteristics.

Suctoreans. Ciliates that are stalked in the adult stage and have rigid tentacles to catch prey.

Supernatant. The liquid remaining above a sediment or precipitate after sedimentation.

Surface overflow rate. A design criterion used in sizing clarifiers. It is typically expressed as the volume of flow per unit amount of clarifier surface area (m^3/m^2 d [gpd/sq ft]).

Suspended solids (SS). lnsoluble solids that either float on the surface of, or are in suspension in, water, wastewater, or other liquids. (2) Solid organic or inorganic particles (colloidal, dispersed, coagulated, flocculated) physically held in suspension by agitation or flow.

Temperature. (1) The thermal state of a substance with respect to its ability to transmit heat to its environment. (2) The measure of the thermal state on some arbitrarily chosen numerical scale such as Celsius or Fahrenheit.

Tertiary treatment. The treatment of wastewater beyond the secondary or biological stage. Tertiary treatment normally implies the removal of nutrients, such as phosphorus and nitrogen, and of a high percentage of suspended solids (see also advanced waste treatment).

Total carbon (TC). A quantitative measure of both total inorganic (TIC) carbon and total organic (TOC) carbon, in milligrams per litre, in water or wastewater, as determined instrumentally by chemical oxidation to CO_2 and subsequent infrared detection in a carbon analyzer.

Total dissolved solids (TDS). The sum of all dissolved solids (volatile and nonvolatile) in wastewater.

Total organic carbon (TOC). The amount of carbon bound in organic compounds in a sample. Because all organic compounds have carbon as the common element, total organic carbon measurements provide a fundamental means of assessing the degree of organic pollution.

Total oxygen demand (TOD). A quantitative measure of all oxidizable material in a sample of wastewater as determined instrumentally by measuring the depletion of oxygen after high-temperature combustion.

Total solids (TS). The sum of dissolved and suspended solids in wastewater.

Total suspended solids (TSS). The amount of insoluble solids floating and in suspension in the wastewater. It is referred to as total nonfilterable residue.

Toxicity. The adverse effect on living organisms by some agent (for example, heavy metals or pesticides).

Trace nutrients. Substances vital to bacterial growth. Trace nutrients are defined in this text as nitrogen, phosphorus, and iron.

Trend analysis. The use of data and statistical tools to study patterns and changes in wastewater treatment processes. Computer software programs aid in the speed and scope of this type of analysis.

Turbidity. (1) A condition in water or wastewater caused by the presence of suspended matter, resulting in the scattering and absorption of light. (2) Any suspended solids imparting a visible haze or cloudiness to water, which can be removed by filtration. (3) An analytical quantity determined by measurements of light scattering and typically reported in turbidity units (Formazin turbidity units (FTU) or Jackson turbidity units (JTU).

Ultimate biochemical oxygen demand (BOD_u). (1) Commonly, the total quantity of oxygen required to satisfy completely the first-stage biochemical oxygen demand (BOD). (2) More strictly, the quantity of oxygen required to satisfy completely both the first- and second-stage BOD_5.

Virus. The smallest lifeform capable of producing infection and disease in humans or other large species.

Volatile solids (VS). Materials, generally organic, that can be driven off from a sample by heating, typically to 550°C (1022°F); nonvolatile inorganic solids (ash) remain.

Volatile suspended solids (VSS). That fraction of suspended solids, including organic matter and volatile inorganic salts, that will ignite and burn when placed in an electric muffle furnace at 550°C (1022°F) for 60 minutes.

Volumetric loading. The amount of flow applied to a treatment process per unit time per unit volume of the basin or clarifier.

Waste sludge. Biological sludge that is drawn off to be conditioned for ultimate disposal (also called waste activated sludge [WAS]; see also excess sludge and return sludge).

Weir. (1) A diversion dam. (2) A device that has a crest and some side containment of known geometric shape, such as a V, trapezoid, or rectangle, and is used to measure flow of liquid. The liquid surface is exposed to the atmosphere. Flow is related to upstream height of water above the crest, position of crest with respect to downstream water surface, and geometry of the weir opening.

Weir overtlow rate. The amount of flow applied to a treatment process (typically a clarifier) per linear measure of weir (m^3/m d [gpd/ft]).

Washout. The condition whereby excessive influent flows (typically at peak flow conditions) cause the solids in the aeration basins and/or clarifiers to be carried over into downstream processes or discharged to the receiving stream.

Appendix E

Tools

A brief description of some used tools in the development of DAI-DEPUR is described in this appendix: the ACSL simulation language, the G2 shell, the Sun Common Lisp language, the LINNEO^{+} unsupervised classification tool and GAR, the automatic rule generator.

ACSL, Advanced Continuous Simulation Language [Mitchell and Gautier, 1987], is a simulation language designed to modelling and evaluating continuous systems that are described by non-linear differential equations with a unique independent variable. The language also supports solving the equations in the form of transference functions. The language provides the user with a big number of easy usable operators oriented to simulation purposes such as delays, pulses, sinusoidal functions, *etc.*, as well as some graphical outputs. ACSL is a software from Mitchell & Gautier Corp., and probably, nowadays is one of the most common used simulation languages in modelling continuous processes [De Prada, 1989]. There are several commercial versions for PCs, great mainframes and workstations. ACSL compiles the source programs of the user to Fortran code. Thus, it allows to directly write Fortran code supporting a rich interoperability between Fortran and ACSL.

G2 [Gensym, 1992; Gensym, 1990] is a tool for developing and running real-time expert systems for complex applications that require continuous and intelligent monitoring, diagnosis, and control. It is designed for a wide range of real-time applications in such diverse fields as process control, aerospace, and finance. G2 combines many technologies including: rule-based reasoning, simulation, procedural computation, network connectivity, data input and output, object-oriented modelling, modularized design system, and graphical user interfaces. G2 provides an integrated environment consisting of developer's tools, a robust language, and hierachies for structuring the applications: items and class hierarchy, modules and module hierarchy (different knowledge bases), workspaces and workspace hierarchy (different rule modules). G2 also has a number of external data interfaces that allow it to interact with other processes and to receive data from external sources. They are easy to configure and work automatically while a knowledge base runs. The four external data inetrfaces are: G2 Standard Interface (GSI), G2 File Interface (GFI), G2-to-G2 Interface and the Foreign Function Interface.

Sun Common Lisp [Sun Microsystems, 1990] is the Common Lisp dialect from Sun Microsystems. It is a functional programming language that provides a programming environment to writing large, complex programs that can manipulate symbols as well as numbers. It supplies additional features to basic Common Lisp as the Foreign Function Interface, the Multitasking Facility, I/O extensions, the delivery tool kit, the window tool kit, *etc.*, and supports the main especifications of Common Lisp such as the package system, the Common Lisp Object Systems (CLOS), *etc.*

The classification tool for ill-structured domains, LINNEO$^+$ [Béjar, 1995; Béjar *et al.*, 1994], and the automatic rule generator tool, GAR [Riaño, 1994], are already detailed in 5.4.1.1.

Bibliography

[Aamodt, 1989] A. Aamodt. Towards robust expert systems that learn from experience. An architectural framework. *Procc. of 3rd European Knowledge Acquisition for Knowledge-Based Systems Workshop (EKAW-89)*, Paris, pp. 311-326, 1989.

[Aamodt and Plaza, 1994] A. Aamodt and E. Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. *AI Communications* 7(1):39-59, 1994.

[Alterman, 1988] R. Alterman. Adaptive planning. *Cognitive Science* 12:393-422, 1988.

[Aguilar, 1990] J. Aguilar. Knowledge-based systems for the supervision of real-time control process. *IV International Symposium on Knowledge Engineering*, Barcelona, 1990.

[Aguilar *et al.*, 1992] J. Aguilar, A. Delgado and J.L. De La Rosa. Principles of Knowledge-based control systems. *V International Symposium on Knowledge Engineering*, Sevilla, pp. 298-303, 1992.

[Aikins, 1983] J. Aikins. Prototypical knowledge for expert systems. *Artificial Intelligence* 20:163-210, 1983.

[Alamán *et al.*, 1992] X. Alamán, S. Romero, C. Aguirre, P. Serrahima, R. Muñoz, V. López, J. Dorronsoro and E. de Pablo. MIP: A Real Time Expert System. *8th Conf. on Artificial Intelligence Applications (CAIA-92)*. Monterrey, California, 1992.

[Alex *et al.*, 1994] J. Alex, U. Jumar and R. Tschepetzki: A Fuzzy Controller for Activated Sludge Wastewater Plants. In *Procc. of the 2nd IFAC/IFIP/IMACS Int. Symp. on Artificial Intelligence in Real Time Control (AIRTC'94)*, pp. 75-80, València, October, 1994.

[Alleman *et al.*, 1992] J.E. Alleman, M.W. Sweeney and D.A. Vaccari (editors). Special issue on Applying Instrumentation and Automation in Environmental Engineering: Water and Wastewater. *ISA Transactions* 31(1), 1992.

[Allen and Langley, 1990] J.A. Allen and P. Langley. Integrating memory and search in planning. In *Proc. of the DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*. San Diego, CA, 1990.

[Allworth and Zobel, 1987] S.T. Allworth and R.N. Zobel. *Introduction to Real-time System Design*. Macmillan, London, 1987.

[Anderson, 1983] J.R. Anderson. *The architecture of Intelligence.* Harvard University Press, 1983.

[Anderson *et al.*, 1990] K. Anderson, D. Coleman, C. Hill, A. Jaworski, P. Love, D. Spindler and M. Simaan. Special Cause Management: a Knowledge-Based approach to Statistical Process Control. *Annals of Mathematics and Artificial Intelligence* 2:21-38, 1990.

[Arcos and Plaza, 1995] J.L. Arcos and E. Plaza. Reflection in NOOS: an Object-centered representation language for knowledge modelling. In *IJCAI Workshop on Reflection and Meta-level architectures and their application in AI (IJCAI'95)*, pages 1-10, Montréal, 1995.

[Ashley, 1990] K.D. Ashley. *Modelling legal argument: reasoning with cases and hypotheticals*. The MIT Press, 1990.

[Åström *et al.*, 1986] K.J. Åström, J.J. Anton and K.E. Årzén. Expert Control. *Automatica* 22:277-286, 1986.

[Baim, 1988] P.W. Baim. A method for attribute selection in interactive learning systems. *IEEE trans. on Pattern Analysis and Machine Intelligence* 10, 1988.

[Bain, 1986] W. Bain. *Case-based reasoning: a computer model of subjective assessment*. Ph. D. Dissertation. Dept. of Computer Science. Yale University, 1986.

[Barachini and Theuretzbacher, 1988] F. Barachini and N. Theuretzbacher. The Challenge of Real-Time Process Control for Production Systems. *Procc. of AAAI Conference (AAAI-88)*, pp. 705-709, 1988.

[Bareiss, 1989] E.R. Bareiss. *Exemplar-based knowledge acquisition: a unified approach to concept representation, classification and learning.* Academic Press, 1989.

[Bañares-Alcántara and Ponton, 1992] R. Bañares-Alcántara and J.W. Ponton. Artificial Intelligence techniques in Chemical Engineering Process

Design. *Applications in Artificial Intelligence in Engineering,* VII, Computational Mechanics Publications, Bath. England, pp. 581-607, 1992.

[Beck, 1986] M.B. Beck. Identification, estimation and control of biological waste-water treatment processes. *IEEE Proceedings* 133:254-264, 1986.

[Beck *et al.*, 1990] M.B. Beck, J.P. Lambers, H.E.C. Mackenzie and P.W. Jowitt. A Prototype Expert System for Operational Control of the Activated Sludge Process. Internal Report. Imperial College, Dept. of Civil Engineering, London. (Previously published in French as Un Prototype de Système Expert pour le Contrôle d'un Procédé de Boues Activées. *Sciences et Techniques de l'Eau* 23 (2):161-167, 1990.

[Beck *et al.*, 1978] M.B. Beck, A. Latten and R.M. Tong. Modelling and Operational Control of the Activated Sludge Process in Wastewater Treatment. Professional Paper PP-78-10, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.

[Becker, 1987] B. Becker. The limits of knowledge acquisition. Report FB-14. Dortmun University, 1987.

[Béjar, 1995] J. Béjar. *Knowledge Acquisition in ill-structured domains.* Ph.D. Thesis. Dept. de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. (In Spanish), 1995.

[Béjar *et al.*, 1994] J. Béjar, U. Cortés and M. Domingo. Using Domain Theory to Bias Classification Processes in Ill-Domains. *Procc. of IV Congreso Iberoamericano de Inteligencia Artificial (IBERAMIA-94),* Caracas, Venezuela, pp. 187-199, 1994.

[Béjar and Cortés, 1992] J. Béjar and U. Cortés (1992). LINNEO$^+$: Herramienta para la adquisición de conocimiento y generación de reglas de clasificación en dominios poco estructurados. *Actas 3er Congreso Iberoamericano de Inteligencia Artificial, (IBERAMIA-92),* La Habana, Cuba, pp. 471-481, 1992.

[Belanche and Cortés, 1991] L. Belanche and U. Cortés. The nought attributes in knowledge-based systems. *Proc. of European Workshop on Verification and Validation (EUROVAV-91),* pp. 77-102. Cambridge, UK, 1991.

[Belanche *et al.*, 1992b] Ll. Belanche, M. Sànchez, U. Cortés and P. Serra. A knowledge-based system for the diagnosis of waste-water treatment plants. *Proc. of 5th Int. Conf. on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-92),* Paderborn (Germany), LNAI-604, Springer Verlag, pp. 324-336, 1992.

[Belanche *et al.*, 1992a] Ll. Belanche, M. Sànchez and U. Cortés. Un sistema basado en el conocimiento para la diagnosis en plantas de tratamiento de aguas residuales. *3er Congreso Iberoamericano de Inteligencia Artificial (IBERAMIA-92)*, La Habana (Cuba), 1992.

[Benefield and Randall, 1980] L.D. Benefield and C.W. Randall. *Biological process design for wastewater treatment*. Prentice-Hall, N.J., 1980.

[Bennett, 1987] S. Bennett. *Real-Time Computer Control: An Introduction*. Prentice-Hall, Hemel Hempstead, 1987.

[Bernard, 1988] J.A. Bernard. Use of a Rule-based System for Process Control. *IEEE Control Systems Magazine*, October, pp. 3-13, 1988.

[Berthuex *et al.*, 1987] P.M. Berthuex, M. Lai and Darjatmoko. A Statistics-based information and expert system for plant control and improvement. Procc. of *5th National Conf. on Microcomputers in Civil Engineering*, (W.E. Carrol, editor), Orlando, Florida, pp. 146-150, 1987.

[Bond, 1989] A.H. Bond. The Cooperation of Experts in Engineering Design. In *Distributed Artificial Intelligence* (L. Gasser and M.N. Huhns editors) Vol. II, Pitman Publishing / Morgan Kaufmann Publishers, Los Altos, CA, pp. 463-484, 1989.

[Bond and Gasser, 1988] A.H. Bond and L. Gasser (editors). *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[Bonissone, 1994] P. Bonissone. Fuzzy logic controllers: an industrial reality. In *Computational Intelligence Imitating Life* (J.M. Zurada, R.J. Marks II and C.J. Robinson, editors). IEEE Press, 1994.

[Bonisone, 1993] P. Bonissone. Knowledge representation and inference in first generation knowledge based systems. In *Advances in Expert Systems for Management* Vol. 1, pp. 59-86, (Wallace and Grabowski, editirs). JAI Press, Greenwich, CT, 1993.

[Bonissone, 1979] P. Bonissone. *The problem of linguistic aproximation in system analysis*. Ph.D. Thesis. EECS Dept., U. C. Berkeley. Univ. Microfilms Intern. Publications #8014618, Ann Arbor, Michigan, 1979.

[Bonissone and Decker, 1985] P. Bonissone and K. Decker. Selecting uncertainty calculi and granularity: an experiment in trading-off precision and complexity. *In Proc. of Workshop on Uncertainty and Probability in Artificial Intelligence*, pp. 57-66, Los Angeles, CA, 1985.

[Bonissone and Tong, 1985] P. Bonissone and R.M. Tong. Editorial: reasoning with uncertainty in expert systems. *Int. Journal of Man-Machine Studies* 22:241-250, 1985.

[Bonissone *et al.*, 1990] P. Bonissone, Sahnoun and DiCesare. Linguistic summarization of fuzzy data. *Journal of Information Science* 52:141-152, 1990.

[Booser and Bredshaw, 1987] J. Booser and J. Bredshaw. AQUINAS: a knowledge acquisition workbench for building knowledge-based systems. *Proc. of Int. Conference on Expert Systems (IASTED)*, 1987.

[Bouslama, 1992] F. Bouslama. Fuzzy Control and their Natural Control Laws. *Fuzzy Sets and Systems* 48:65-86, 1992.

[Brajnik, 1989] G. Brajnik. Epistemology, Organization and Use of Functional Knowledge for Reasoning about Physical Systems. *Procc. of 10th Int. Workshop on Experts Systems and their Applications*, pp. 53-66, 1989.

[Brandau *et al.*, 1991] R. Brandau, A. Lemmon and C. Lafond. Experience with extended episodes: cases with complex temporal structure. In *Proc. of Workshop on case-based reasoning (DARPA)*. Washington D.C., 1991.

[Branting, 1991] L.K. Branting. Exploiting the complemetarity of rules and precedents with reciprocity and fairness. In *Proc. of Case-Based Reasoning Workshop,* pp. 39-50. Morgan Kaufmann. May, 1991.

[Buchanan and Shortlife, 1984] B.G. Buchanan and E.H. Shortlife. *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming project*. Addison-Wesley, 1984.

[Buchanan and Smith, 1988] B. Buchanan and R. Smith. Fundamentals of expert systems. *Annual Reviews on Computer Science* 3:23-58, 1988.

[Burns and Wellings, 1990] A. Burns and A. Wellings. Real-Time Systems and their Programming Languages. Addison-Wesley Publishing Company, 1990.

[Cammarata *et al.*, 1983] S. Cammarata, D. McArthur and R. Steeb. Strategies of Cooperation in Distributed problem Solving. *Procc. 8th Int. Joint Conference on Artificial Intelligence,* Karlsruhe, West Germany, pp. 279-284, 1983.

[Capodaglio *et al.*, 1991] A.G. Capodaglio, H.V.Jones, V. Novotny and X. Feng. Sludge bulking analysis and forecasting: application of system identification and artificial neural computing technologies. *Water Research* 25(10):1217-1224, 1991.

[Carbonell, 1986] J. Carbonell. Derivational analogy: a theory of reconstructive problem solving and expertise acquisition. *Machine Learning* vol. 2, 1986.

[Carbonell *et al.*, 1990] J. Carbonell, C. Knoblock and S. Minton. PRODIGY: an integrated architecture for planning and learning. In *Architectures for intelligence* (K. Van Lehn, editor). Erlbaum, 1990.

[Castillo and Quintanilla, 1991] L. Castillo and G. Quintanilla. Initial concepts in Distributed Artificial Intelligence. In Proc. of *Approaches to non-conventional Computing: towards Intelligent Systems (TEC-COMP 91),* pp. 76-97, 1991.

[Cavanna *et al.*, 1989] A. Cavanna, J.C. Chautard, C. Honorat, M. Levin and B. Klausen. QUIC Toolkit Demonstrator Applications. *CIM Europe*, 1989.

[Chaib-draa *et al.*, 1992] B. Chaib-draa, R. Mandiau and P. Millot. Distributed Artificial Intelligence: An Annotated Bibliography. *SIGART Bulletin* 3(3):20-37, 1992.

[Chandrasekaran, 1983] B. Chandrasekaran. Towards a taxonomy of problem-solving types. *AI Magazine* 7(3):66-77, 1983.

[Charniak and McDermott, 1985] E. Charniak and D. McDermott. *Introduction to Artificial Intelligence*. Addison-Wesley, 1985.

[Charniak *et al.*, 1987] E. Charniak, C.K. Riesbeck, D.V. McDermott and J.R. Meehan. *Artificial Intelligence Programming*. Second edition, Lawrence Erlbaum Associates, 1987.

[Cheeseman *et al.*, 1988] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor and D. Freeman. AUTOCLASS: a Bayesian Classification System. *Procc. of 5th Int. Conference on Machine Learning (ICML- 1988)*, pp. 54-64. Ann Arbor, MI, 1988.

[Chudoba, 1985] J, Chudoba. Control of activated sludge filamentous bulking: formulation of basic principles. *Water Research* 19:1017-1022. 1985.

[Clancey, 1985b] W. Clancey. Heuristic Classification. *Artificial Intelligence* 27(3):289-350, 1985.

[Clancey, 1985a] W. Clancey. The epistemology of rule-based systems: a framework for explanation. *Artificial Intelligence* 20(3):215-251, 1985.

[Clarke *et al.*, 1987b] D.W. Clarke, C. Mothad and P.S. Tuffs. Generalized Predictive Control–Part II: Extensions and Interpretations. *Automatica* 23:149-160, 1987.

[Clarke *et al.*, 1987a] D.W. Clarke, C. Mothad and P.S. Tuffs. Generalized Predictive Control–Part I: the Basic Algorithm. *Automatica* 23:137-148, 1987.

[Cognitive, 1992] Cognitive Systems. *ReMind Developer's Reference Manual*. Boston, 1992.

[Corder and Lee, 1986] G.D. Corder and P.L. Lee. Feedforward control of a wastewater plant. *Water Research* 20:301-309, 1986.

[Cuda *et al.*, 1994] T.V. Cuda, J.P. Baukus, R.L. Seliger and D. Chow. Multi-paradigm reasoning for molecular beam epitaxi control. *2nd IFAC Workshop on Computer Software Structures Integrating AI/KBS (CSI-AI/KBS'94)*, Lund (Sweden), pp. 65-70, 1994.

[Culp and Culp, 1978] R.L. Culp and G.L. Culp. *Handbook of Advanced Wastewater Treatment*. 3rd edition. Van Nostrand-Reinhold Co., New York, 1978.

[Czoagala and Rawlik, 1989] E. Czoagala and T. Rawlik. Modelling of a Fuzzy Controller with application to the Control of Biological Processes. *Fuzzy Sets and Systems* 31:13-22, 1989.

[Danyluk, 1987] A.P. Danyluk. The Use of Explanations for Similarity-based Learning. *Procc. of International Joint Conference on Artificial Intelligence (IJCAI-87)*, pp. 274-276, 1987.

[Davis, 1980] R. Davis. Reasoning about control. *Artificial Intelligence* 15(3):179-222, 190.

[Davis, 1979] R. Davis. Interactive transfer of expertise. *Artificial Intelligence* 12:121-157, 1979.

[Davis and Lenat, 1982] R. Davis and D.B. Lenat, editors. *Knowledge-Based Systems in Artificial Intelligence*. McGraw-Hill, 1982.

[De Gràcia, 1993] J. De Gràcia. *Classification Techniques Evaluation for bioprocesses management: Application to an activated sludge reactor*. Master's Thesis. Dept. de Química. Universitat Autonòma de Barcelona. (In Catalan), 1993.

[Dellepiane *et al.*, 1992] S.G. Dellepiane, G. Venturi and G.L. Vernazza. Model generation and model matching of medical images by a fuzzy approach. *Pattern Recognition* 25(2):115-137, 1992.

[De Prada, 1989] C. De Prada. *Modelado y simulación en control de procesos. Tècniques de control i optimització de processos per ordinador*. Notes of the European Social Funding–Institut de Ciències de l'Educació (I.C.E.–U.A.B.) Course. Bellaterra, 1989.

[De Prada *et al.*, 1991] C. De Prada, R, Moreno, M. Poch and J. Robusté. Recursive estimation of OUR in activated sludge process. In *European Control Conference (ECC-91)*. Grenoble (France), 1991

[Dietterich and Michalsky, 1981] T.G. Dietterich and R.S. Michalsky. Inductive Learning of Structural Descriptions: Evaluation Criteria and Comparative Review of Selected Methods. *Artificial Intelligence* 16:257-294, 1981.

[Dijkstra, 1975] E.W. Dijkstra. "Guarded Commands, Nondeterminacy and Formal Derivation of Programs. *Communications of the ACM* 18(8). August, 1975.

[Dochain, 1991] D. Dochain. Design of adaptive controllers for non-linear stirred tank bioreactors: extension to the MIMO situation. *Journal of Process Control* 1:41-48, 1991.

[Dubes and Jain, 1990] R. Dubes and A. Jain. *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, USA, 1990.

[Dutta and Bonissone, 1993] S. Dutta and P. P. Bonissone. Integrating case-based reasoning and rule-based reasoning. *Journal of Approximate Reasoning* 8(3):163-203, 1993.

[Dutta and Bonissone, 1991] S. Dutta and P. P. Bonissone. Integrating case-based and rule-based reasoning: the possibilistic connection. In *Uncertainty in Artificial Intelligence*, Vol. 6 (P. Bonissone, M. Henrion, L.N. Kanal, and J. Lemmer, editors), pp. 281-298. North Holland, 1991.

[Dym and Levitt, 1991] C.L. Dym and R.E. Levitt. *Knowledge-based Systems in Engineering*. McGraw-Hill, 1991.

[Efstathiou and Mamdani, 1985] J. Efstathiou and A. Mamdani (editors). *Expert Systems and Optimization in Process Control*. Technical Press, 1985.

[Finch *et al.*, 1990] F.E. Finch, O.O. Oyeleye and M.A. Kramern. A Robust Event-Oriented Methodology for Diagnosis of Dynamic Process Systems. *Computers & Chemical Engineering* 14(12):1379-1396, 1990.

[Fisher, 1987] D. Fisher. Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning* 2:139-172, 1987.

[Fisher and Pazzani, 1991] D.H. Fisher and M.J. Pazzani. Computational models of concept learning. In *Concept Formation: Knowledge and Experience in Unsupervised Learning* edited by D.H. Fisher, M.J. Pazzani and P. Langley. Morgan Kaufmann. San Mateo, CA, 1991.

[Flanagan, 1980] M.J. Flanagan. On the Application of Approximate Resaoning to the Control of Activated Sludge Process. *Procc. of Joint Automatic Control Conference*, ASME, San Francisco, CA, 1980.

[Flanagan, 1979] M.J. Flanagan. Upgrading the Activated Sludge Process through Automatic Control. *American Institute of Chemical Engineers, Symposium Series* 75(190):232-242, 1979.

[Fox and Leake, 1995] S. Fox and D.B. Leake. Using introspective reasoning to refine indexing. *Proc. of 14th Int. Joint Conference on Artificial Intelligence (IJCAI'95)*, pp. 391-397. Montréal, 1995.

[Fu, 1971] K.S. Fu. Learning Control Systems and Intelligent Control Systems: an Intersection of Artificial Intelligence and Automatic Control. *IEEE Trans. on Automatic Control* 16(1), 1971.

[Fum *et al.*, 1988] D. Fum, G. Guida and C. Tasso. Distributed Multi-Agent Architecture for Natural Language Processing. *Procc. of Int. Conf. on Computational Linguistics*, Budapest (Hungary), pp. 812-814, 1988.

[Gall and Patry, 1989] R. Gall and G. Patry. Knowledge-based system for the diagnosis of an activated sludge plant. In *Dynamic Modelling and Expert Systems in Wastewater Engineering*. (G. Patry and D. Chapman editors), Chelsea, MI. Lewis Publishers, 1989.

[Gasser, 1987] L. Gasser. Pressure: an Adaptive, Distributed, Multi-Robot Task Allocation System. In *Procc. of the SME Conf. on AI in Manufacturing*, Long Beach, CA, 1987.

[Geldof *et al.*, 1993] S. Geldof, L. Steels and W. Van de Welde. COMMET for building knowledge systems. *CC-AI* 10(3):195-218, 1993.

[Genesereth, 1983] M.R. Genesereth. An overview of meta-level architecture. In *Proc. of National Conference on Artificial Intelligence (AAAI-83)*, pp. 119-124, 1983.

[Gennari *et al.*, 1989] J.H. Gennari, P. Langley and D. Fisher. Model of Incremental Concept Formation. *Artificial Intelligence* 40(1-3):11-61, 1989.

[Gensym, 1992] Gensym. *G2 Reference Manual, version 3.0.* Gensym Corporation. Cambridge, MA. July, 1992.

[Gensym, 1990] Gensym. *G2 Reference Manual.* Gensym Corporation, Cambridge, MA. January, 1990.

[Gil, 1991]. Y. Gil. Integrated architectures for Artificial Intelligence. In *Proc. of Approaches to non-conventional Computing: towards Intelligent Systems (TEC-COMP 91)*, pp. 41-55, 1991.

[Goel and Chandrasekaran, 1992] A. Goel and B. Chandrasekaran, Case-based design: a task analysis. In *Artificial Intelligence approaches to Engineering design, vol. 2: Innovative design* (C. Tong and D. Sriram, editors). Academic Press, 1992.

[Golding and Rosenbloom, 1991] A.R. Golding and P.S. Rosenbloom. Improving rule-based systems through case-based reasoning. In *Proc. of National Conference on Artificial Intelligence (AAAI-91)*, pp. 22-27. AAAI Press. August, 1991.

[Gómez and Chandrasekaran, 1981] F. Gómez and B. Chandrasekaran. Knowledge Organization and Distribution for Medical Diagnosis. In *IEEE Transactions on Systems, Man and Cybernetics* 11(1):34-42, 1981.

[González and Dankel, 1994] González and Dankel. *The engineering of knowledge-based systems.* Prentice-Hall, 1994.

[Hammond, 1989] K. Hammond. *Case-based planning: viewing planning as a memory task.* Academic Press, 1989.

[Hanson and Bauer, 1986] S.T. Hanson and M. Bauer. Conceptual Clustering, Categorization and Polymorphy. *Machine Learning* 1:343-372, 1986.

[Hart, 1984] P. Hart. Artificial Intelligence in transistion. In *Knowledge-based problem solving* (J. Kowalik, editor), pp. 296-311, Prentice-Hall, 1984.

[Hayes-Roth, 1984] F. Hayes-Roth. The knowledge-based expert system: a tutorial. *IEEE Computer*, pp. 11-28, September, 1984.

[Hayes-Roth and McDermott, 1977] F. Hayes-Roth and J. McDermott. Knowledge Acquisition from structural descriptions. *Procc. of the Int. Joint Conference on Artificial Intelligence (IJCAI-77)*, pp. 356-362. Cambridge, MA, 1977.

[Hennessy and Hinkle, 1992] D.H. Hennessy and D. Hinkle. Applying case-based reasoning to autoclave loading. *IEEE Expert* 7(5):21-26, 1992.

[Henze *et al.*, 1993] M. Henze, R. Dupont, P. Grau and A. de la Sota. Rising sludge in secondary settlers due to denitrification. *Water Research* 27(2):231-236, 1993.

[Hewitt, 1986] C.E. Hewitt. Offices are open systems. *ACM Trans. on Office Information Systems* 4(3):271-287, 1986.

[Hewitt, 1985] C.E. Hewitt. The challenge of open systems. *Byte* 10(4):223-242, 1985.

[Hinrichs, 1992] T.R. Hinrichs. *Problem solving in open worlds: a case study in design*. Lawrence Erlbaum, 1992.

[Hoare, 1978] C.A.R. Hoare. Communicating Sequential Processes. *Communications of the ACM* 21(8):666-677, August, 1978.

[Holland *et al.*, 1986] J.H. Holland, K.J. Holyoak, R.E. Nisbett and P.R. Tagart. *Induction: processes of inference, learning and discovery*. The MIT Press, 1986.

[Horan and Eccles, 1986] N.J. Horan and C.R. Eccles. The Potential for Expert Systems in the Operation and Control of Activated Sludge Plants. *Process Biochemistry*, pp. 81-85, 1986.

[Huang *et al.*, 1991] Y.L. Huang, G. Sundar and L.T. Fan. Min-Cyanide: an expert system for cyanide waste minimization in electroplating plants. *Environmental progress* 10(2):89-95, 1991.

[Huhns, 1987] M.N. Huhns (editor). *Distributed Artificial Intelligence*. Pitman Publishing / Morgan Kaufmann Publishers, San Mateo, CA, 1987.

[Hunt *et al.*, 1992] K.J. Hunt, D. Sbarbaro, R. Zbikowski and P.J. Gawthrop. Neural Networks for Control Systems–A Survey. *Automatica* 28(6):1083-1112, 1992.

[Hushon, 1987] J.M. Hushon. Expert Systems for Environmental Problem. *Environmental Science & Technology* 21(9):838-841, 1987.

[Hydromantis, 1995] Hydromantis, Inc. *GPS-X User's Guide*. Hydromantis Inc., Hamilton, Ontario, Canada, 1995.

[Intellicorp, 1986] Intellicorp. Model-based reasoning in the KEE and SimKit Systems. *Intellinews* 2(2), August, 1986.

[Jackson, 1990] P. Jackson. *Introduction to expert systems*. Addison Wesley, 2nd edition, 1990.

[Jenkins *et al.*, 1993] D. Jenkins, M.G. Richard and G.T. Daigger. *Manual on the causes of control of activated sludge bulking and foaming*. Lewis Publishers, 1993.

[Junta de Sanejament, 1995] Junta de Sanejament. *Programa de sanejament d'aigües residuals urbanes de Catalunya*. Junta de Sanejament. Departament de Medi Ambient. Generalitat de Catalunya, 1995.

[Karr, 1991] C.L. Karr. Genetic Algorithms for Fuzzy Controlers. *AI Expert*, February, 1991.

[Karr et *al.*, 1989] C.L. Karr, L.M. Freeman and D.L. Meredith. Improved Fuzzy Process Control of Spacecraft Autonomous Rendezvous using Genetic Algorithm. Procc. of *SPIE Intelligent and Adaptive Systems Conference*, Philadelphia, pp. 274-288, 1989.

[Kass and Leake, 1988] A.M. Kass and D.B. Leake. Case-based reasoning applied to constructing explanations. *Proc. of Workshop on case-based reasoning (DARPA)*. Clearwater, Florida. 1988.

[Ke and Ali, 1991] M. Ke and M. Ali. Induction in Database Systems: a Bibliography. *Applied Intelligence* 1(3):263-270, 1991.

[Kirn and Schneider, 1992] S. Kirn and J. Schneider. STRICT: Selecting The “RIght” architeCTure. Proc. of *5th Int. Conf. on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-92),* Paderborn, Germany, LNAI-604, Springer-Verlag, pp. 391-400, 1992.

[Ko *et al.*, 1982] K.Y.-J. Ko, B. C. McInnis and G. C. Goodwin. Adaptive control and Identification of the dissolved oxygen process. *Automatica* 18:727-730, 1982.

[Kodukula, 1988] P.S. Kodukula. Expert Systems in Environmental Technology. *Internal report*, Central Research and Engineering Technology of Union Carbide Inc, 1988.

[Kolodner, 1993] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.

[Kolodner, 1985] J.L. Kolodner. Memory for experience. In *The psychology of learning and motivation vol. 19* (G. Bower editor). Academic Press, 1985.

[Kolodner and Simpson, 1989] J.L. Kolodner and R.L. Simpson. The MEDIATOR: analysis of an early case-based problem solver. *Cognitive Science* 13(4):507-549, 1989.

[Kononenko and Bratko, 1991] I. Kononenko and I. Bratko. Information-Based Evaluation Criterion for Classifiers' Performance. *Machine Learning* 6(1):67-80, 1991.

[Kosko, 1992] B. Kosko. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice Hall, USA, 1992.

[Koton, 1989] P. Koton. *Using experience in learning and problem solving*. Ph. D. Thesis. Dept. of Computer Science. MIT, 1989.

[Kraft *et al.*, 1992] L.G. Kraft, W.T. Miller and D. Dietz. Development and Application of CMAC Neural Network-Based Control. *Handbook of Intelligence Control*, (D.A. White and D.A. Sofge, editors) Van Nostrand Reinhold, New York, 1992.

[Krichten *et al.*, 1991] D.J. Krichten, K.D. Wilson and K.D. Tracy. Expert Systems guide biological phosphorus removal. *Water Environmental Technology* pp. 60-64, 1991.

[Krovvvidy and Wee, 1993] S. Krovvidy and W.G. Wee. Wastewater Treatment Systems from Case-Based Reasoning. *Machine Learning* 10:341-363, 1993.

[Krovvidy *et al.*, 1991] S. Krovvidy, W.G. Wee, R. S. Summers and J.J. Coleman. An AI Approach for Wastewater treatment systems. *Applied Intelligence* 1(3):247-261, 1991.

[Kuhn, 1971] T.S. Kuhn. *La Estructura de las Revoluciones Científicas*. Fondo de Cultura Económica, 1971.

[Laffey *et al.*, 1988] T.J. Laffey, P.A. Cox, J.L. Schmidt, S.M. Kao and J.Y. Read. Real-time Knowledge-based Systems. *AI Magazine* 9(1):27-45, 1988.

[Laird *et al.*, 1987] J. Laird, A. Newell and P. Rosenbloom. SOAR: an architecture for general intelligence. *Artificial Intelligence* 33:1-64, 1987.

[Laird *et al.*, 1986] J. Laird, P. Rosenbloom and A. Newell. Chunking in SOAR: the anatomy of a general learning mechanism. *Machine Learning* 1(1):11-46, 1986.

[Lapointe *et al.*, 1989] J. Lapointe, B. Marcos, M. Veillette, G. Laflamme and M. Dumontier. Bioexpert – an Expert System for Wastewater Treatment process diagnosis. *Computers & Chemical Engineering* 13(6):619-630, 1989.

[Lebowitz, 1987] J. Lebowitz. Experiments with Incremental Concept Formation: UNIMEM. *Machine Learning* 2:103-138, 1987.

[Lenat, 1984] D.B. Lenat. The Role of Heuristics in Learning by Discovery: Three Case Studies. In R.S. Michalsky, J.G. Carbonell and T.M. Mitchell editors, *Machine Learning: an Artificial Intelligence Approach*. Springer-Verlag, Berlin, 1984.

[Lenat, 1982] D.B. Lenat. AM: an Artificial Intelligence approach to discovery in Mathematics as heuristic search. In *Knowledge-Based Systems in Artificial Intelligence* (R. Davis and D.B. Lenat, editors). McGraw-Hill, 1982.

[Lenat and Brown, 1983] D.B. Lenat and J.S. Brown. Why AM and Eurisko appear to work. In *Proc. of the National Conference on Artificial Intelligence (AAAI-83)*, 1983.

[Lenat and Guha, 1990] D.B. Lenat and R.V. Guha. *Building large Knowledge-Based systems*. Addison-Wesley, 1990.

[Lesser and Corkill, 1987] V.R. Lesser and D.D. Corkill. Distributed problem solving. In *Encyclopedia of Artificial Intelligence* (S.C. Shapiro editor), pp. 245-251, John Wiley & Sonas, 1987.

[López, 1993] B. López. Learning and plan generation for expert systems. Ph.D. Thesis. Dept. de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya, 1993.

[López de Mántaras, 1991] R. López de Mántaras. A Distance-Based Attribute Selection Measure for Decision Tree Induction. *Machine Learning* 6(1):81-92, 1991.

[López de Mántaras, 1990] R. López de Mántaras. *Approximate reasoning models*. Ellis Horwood Series in Artificial Intelligence. U.K., 1990.

[Maeda, 1989] K. Maeda. A knowledge-based system for the wastewater treatment plant. *Future Generation Computer Systems* 5:29-32. North Holland, 1989.

[Maeda, 1985] K. Maeda. An Intelligent Decision Support System for Activated Sludge Wastewater Treatment Processes. *Instrumentation and control of water and wastewater treatment and transport systems*. Drake editor (IAWPRC) Pergamon Press, 1985.

[Maes, 1988] P. Maes. Issues in computational reflection. In *Meta-level architectures and reflection*, pp. 21-35 (P. Maes and D. Nardi, editors). Elsevier Science Publishers, 1988.

[Maes and Nardi, 1988] P. Maes and D. Nardi, editors. *Meta-level architectures and reflection*. Elsevier Science Publishers, 1988.

[Maler, 1992] O. Maler. Hybrid Systems and Real World Computations. *Workshop on Theory of Hybrid Systems*, Technical University of Denmark, 1992.

[Manna and Pnueli, 1991] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specifications*. Springer-Verlag, New York, 1991.

[Marsilli-Libeli, 1989] S. Marsilli-Libeli. Modelling, identification and control of the activated sludge process. *Advances in Biochemical Engineering* 89, 1989.

[Marsilli-Libeli, 1982] S. Marsilli-Libeli. Optimal control strategies for biological wastewater treatment. *Environmental Systems Analysis and Management.*, S. Rinaldi (editor). North-Holland Publishing Co., pp. 279-287, 1982.

[McDermott, 1988] J. McDermott. A taxonomy of problem-solving methods. In *Automating Knowledge acquisition for expert systems* (S. Marcus, editor), pp. 225-256, Kluwer, 1988.

[Metcalf & Eddy, 1991] Metcalf & Eddy Inc. *Wastewater engineering: treatment / disposal / reuse*. Mc Graw-Hill, 3rd edition, 1991.

[Meystel, 1985b] A. Meystel. Intelligent Control: Issues and Perpectives. *IEEE Worksop on Intelligent Control*, pp. 1-15, 1985.

[Meystel, 1985a] A. Meystel. King Sun Fu: a Life Devoted to the New Frontier of Science. *IEEE Worksop on Intelligent Control*, pp. iii-vi, 1985.

[Michalsky and Reinke, 1986] R.S. Michalsky and R.E. Reinke. Incremental Learning of decision rules: a method and experimental results. *Machine Intelligence* 11. Oxford University Press. Oxford, UK, 1986.

[Michalsky and Stepp, 1984] R.S. Michalsky and R.E. Stepp. Learning from observation: Conceptual Clustering. In R.S. Michalsky, J.G. Carbonell and T.M. Mitchell editors, *Machine Learning: an Artificial Intelligence Approach*. Springer-Verlag, Berlin, 1984.

[Michalsky and Tecuci, 1991] R.S. Michalsky and G. Tecuci (editors). *Proc. of In. Workshop on Multistrategy Learning*. Harpers Ferry, 1991.

[Mitchell, 1990] T.M. Mitchell. Becoming increasingly reactive. In *Proc. of 8th National Conference on Artificial Intelligent (AAAI-90)*. Boston, 1990.

[Mitchell, 1982] T.M. Mitchell. Generalization as Search. *Artificial Intelligence* 18:203-226, 1982.

[Mitchell & Gautier, 1987] Mitchell & Gautier Co. ACSL advanced continuous simulation language. Massachusets, USA, 1987.

[Mitchell *et al.*, 1990] T.M. Mitchell, J. Allen, P. Chalasani, J. Cheng, O. Etzioni, M. Ringuette and J. Schlimmer. Theo: a framework for self-improving systems. In *Architectures for intelligence* (K. Van Lehn, editor). Erlbaum, 1990.

[Miyashita and Sycara, 1995] K. Miyashita and K. Sycara. Improving sytem performance in case-based iterative optimization through knowledge filtering. *Proc. of 14th Int. Joint Conference on Artificial Intelligence (IJCAI'95)*, pp. 371-376. Montréal, 1995.

[Moore *et al.*, 1984] R.L. Moore, L.B. Hawkinson, C.G. Knickerbocker and L.M. Churchman. A Real Time Expert System for Process Control. *IEEE Conference on Artificial Intelligence for Applications (CAIA-84)*, pp. 569-576, 1984.

[Moreno, 1991] R. Moreno. *Non-linear predictive control in aerobic bioreactors*. Master's Thesis. Universitat Autònoma de Barcelona. (In Spanish), 1991.

[Moreno *et al.*, 1992] R. Moreno, C. de Prada, J. Lafuente, M. Poch and G. Montague. Non-linear predictive control of dissolved oxygen in the activated sludge process. *ICCAFT 5 / IFAC-BIO 2 Conference*, Keystone (CO), USA, 1992.

[Moro, 1984] P.P. Moro. Métodos biológicos para el control de los procesos de plantas de tratamiento de aguas industriales. *Tecnología del Agua* 13:41-47, 1984.

[Motus, 1994] L. Motus. Trends In Artificial Intelligence Applications for Real-Time Control (a Speculative Study). *2nd IFAC/IFIP/IMACS Symposium on Artificial Intelligence in Real Time Control (AIRTC'94)*. pp. 36-45. València, 1994.

[Motus, 1990] L. Motus. *Dynamics of Embedded Software*. Valgus, Tallinn (Estonia), 1990.

[Narendra and Parthasarathy, 1990] K. S. Narendra and K. Parthasarathy. Identification and Control of Dynamic Systems using Neural Networks. *IEEE Trans. on Neural Networks* 1(1):4-27, 1990.

[Neches *et al.*, 1991] R. Neches, R. Fikes, T. Finin, T. Gruber, T. Senator and W. Swartout. Enabling technology for knowledge sharing. *AI Magazine* 12:36-56, 1991.

[Newell, 1990] A. Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA, 1990.

[Newell, 1982] A. Newell. The Knowledge Level. *Artificial Intelligence* 18(1):87-127, 1982.

[Newell and Simon, 1976] A. Newell and H.A. Simon. Computer Science as Empirical Inquiry: Symbols and search. *Communications of the ACM* 19(3):113-126, 1976.

[Nirenburg and Lesser, 1988] S. Nirenburg and V. Lesser. Providing Intelligent Assistance in Distributed Office Environments. In *Readings in Distributed Artificial Intelligence* (A.H. Bond and L. Gasser editors), Morgan Kaufmann, San Mateo, CA, pp. 590-598, 1988.

[Novotny *et al.*, 1990] V. Novotny, H. Jones, X. Feng and A.G. Capodaglio. Time Series Analysis Models of Activated Sludge Plants. *Water Science & Technology* 23(4-6):1107-1116, 1990.

[Olsson and Andrews, 1977] G. Olsson and J. F. Andrews. The Dissolved Oxygen Profile–a valuable Tool for Control of the Activated Sludge Process. *Water Research* 12:985-1004, 1977.

[Patry and Chapman, 1989] G. Patry and D. Chapman (editors). *Dynamic Modelling and Expert Systems in Wastewater Engineering*. Chelsea, MI. Lewis Publishers, 1989.

[Piskunov, 1992] A. Piskunov. Fuzzy Implication in Fuzzy Systems Control. *Fuzzy Sets and Systems* 45:25-35, 1992.

[Plaza, 1987] E. Plaza. *EAR: a knowledge acquisition and structuration aided system*. Ph.D. Thesis. Dept. de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. (In Catalan), 1987.

[Plaza *et al.*, 1993] E. Plaza, A. Aamodt, A. Ram, W. Van de Velde and M. Van Someren. Integrated Learning Architectures. *Procc. European Conference on Machine Learning (ECML-93),* LNAI-667, Springer-Verlag, pp. 429-441, 1993.

[Polianova, 1994] R. Polianova. Using KREST for expert sytem development. AI-MEMO 94-02, AI Lab, Vrije Universiteit Brussel, Brussels.

[Porter *et al.*, 1990] B.W. Porter, R. Bareiss and R.C. Holte. Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence* 45:229-263, 1990.

[Pujol *et al.*, 1990] R. Pujol, A. Vachon and G. Martin. Guide technique sur le foisonnement des boues activées. Centre National du Machinisme Agricole du Génie Rural des eaux et des fôrets, 1990.

[Puyol, 1994] J. Puyol. *Modularization, Uncertainty, Reflective Control and Deduction by Specialization in MILORD II: a language for Knowledge-Based Systems*. Ph.D. Thesis. Universitat Autònoma de Barcelona, 1994.

[Quinlan, 1986] J.R. Quinlan. Induction of Decision Trees. *Machine Learning* 1:81-106. Kluwer Academic Publishers. Boston, MA, USA, 1986.

[Ram, 1993] A. Ram. Indexing, elaboration and refinement: incremental learning of explanatory cases. *Machine Learning* 10(3):201-248, 1993.

[Ram and Hunter, 1992] A. Ram and L. Hunter. The use of explicit goals for knowledge to guide inference and learning. *Applied Intelligence* 2(1):47-73, 1992.

[Ramparany, 1994] F. Ramparany. Model-based Control with large Temporal Delays. *2nd IFAC/IFIP/IMACS Symposium on Artificial Intelligence in Real Time Control (AIRTC'94)*. pp. 473-478. València, 1994.

[Rao, 1992] M. Rao. *Integrated Systems for Intelligent Control*. Springer-Verlag, 1992.

[Redmond, 1992] M.A. Redmond. Learning by observing and understanding expert problem solving. Georgia Institute of Technology. College of Computing. Technical report GIT-CC-92/43, 1992.

[Riaño, 1994] D. Riaño. *Automatic Knowledge Generation from Data in Classification Domains*. Master's Thesis. Dept. de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya, 1994.

[Rich and Knight, 1991] E. Rich and K. Knight. *Artificial Intelligence*. McGraw-Hill, 2nd edition, 1991.

[Rich and Venkatasubramanian, 1987] S.H. Rich and V. Venkatasubramanian. Model-based reasoning in diagnostic expert systems for chemical process plants. *Computers & Chemical Engineering* 11:111-122, 1987.

[Riesbeck and Schank, 1989] C.K. Riesbeck and R.S. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates Publishers, 1989.

[Rissland and Skalak, 1989] E.L. Rissland and D.B. Skalak. Combining case-based and rule-based reasoning: a heuristic approach. In *Proc. of 11th Int. Joint Conference on Artificial Intelligence (IJCAI-89)*, pp. 524-530. Morgan Kaufmann, August, 1989.

[Robusté, 1990] J. Robusté. *Modelling and Identification of the Activated Sludge Process*. Ph. D. Thesis. Departament de Química. Universitat Autònoma de Barcelona. (In Catalan), 1990.

[Roda *et al.*, 1990] C. Roda, N. Jennings and E.H. Mamdani. A Cooperation Framework for Industrial Process Control. In *Procc. of the Int. Conf. on Cooperating Knowledge Base Systems*, Keele, England, pp. 54-57, 1990.

[R.-Roda, 1994] I. R.-Roda. *Knowledge-based system definition for urban wastewater treatment plants control with nitrification/denitrification criteria.* Master's Thesis. Dept. de Química. Universitat Autonòma de Barcelona. (In Catalan), 1994.

[Russell and Norvig, 1995] S. Russell and P. Norvig. *Artificial Intelligence. A modern approach*. Prentice Hall, 1995.

[Russell and Wefald, 1991] S. Russell and E. Wefald. *Do the right thing. Studies in limited rationality*. The MIT Press, 1991.

[Sacerdoti, 1977] E.D. Sacerdoti. *A structure for plans and behavior*. North-Holland, 1977.

[Sànchez *et al.*, 1995e] M. Sànchez, U. Cortés, J. De Gràcia, J. Lafuente and M. Poch. Concept Formation in WWTP by means of Classification Techniques: a Compared Study. To appear in *Applied Intelligence*, Accepted in December, 1995.

[Sànchez *et al.*, 1995d] M. Sànchez, U. Cortés, I. R.-Roda, M. Poch and J. Lafuente. Learning and adaptation in WWTP through Case-Based reasoning. Submitted to an special issue on Machine Learning of *Microcomputers in Civil Engineering*. December, 1995.

[Sànchez *et al.*, 1995c] M. Sànchez, U. Cortés, J. Lafuente, I. R.-Roda and M. Poch. Knowledge-based techniques in wastewater treatment plants management. Submitted to *Enciclopaedia of Life Support Systems (EOLSS).* September, 1995.

[Sànchez *et al.*, 1995b] M. Sànchez, U. Cortés, I. R.-Roda and M. Poch. Integrating general expert knowledge and specific experimental knowledge in WWTP. *IJCAI'95 Workshop on Artificial Intelligence and the Environment*, pp. 75-81. Montréal, 1995.

[Sànchez *et al.*, 1995a][1] M. Sànchez, U. Cortés, J. Lafuente, I. R.-Roda and M. Poch. DAI-DEPUR: an integrated and distributed architecture for wastewater treatment plants supervision. Submitted to *Artificial Intelligence in Engineering*. March, 1995.

[Sànchez *et al.*, 1994b] M. Sànchez, I. R-Roda, J. Lafuente, U. Cortés and M. Poch. DAI-DEPUR Architecture: Distributed Agents for Real-Time WWTP Supervision and Control. *2nd IFAC/IFIP/IMACS Symposium on Artificial Intelligence in Real Time Control (AIRTC'94)*, València, pp. 179-184, 1994.

[Sànchez *et al.*, 1994a] M. Sànchez, I. R.-Roda, J. Lafuente, U. Cortés and M. Poch. Real Time Supervision of Wastewater Treatment Plants: a Distributed Approach. *2nd IFAC Workshop on Computer Software Structures Integrating AI/KBS (CSI-AI/KBS'94)*, Lund (Sweden), pp. 188-193, 1994.

[Sànchez, 1991] M. Sànchez. *DEPUR: Application of Knowledge-based Systems to diagnosis in wastewater treatment plants*. Master's Thesis. Research Report LSI-91-35. Dept. de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. (In Catalan), 1991.

[Sanz *et al.*, 1989] R. Sanz, J. Aguilar, C. Sierra, Ll. Godó and A. Ollero. Adaptive control with a supervisor level using a rule-based inference system with approximate reasoning. *Artificial Intelligence in Scientific Computation: towards 2nd generation systems*. IMACS, 1989.

[Saridis, 1985] G.N. Saridis. Foundations of the Theory of Intelligent Control. *IEEE Worksop on Intelligent Control*, 1985.

[Schank, 1982] R. Schank. *Dynamic memory: a theory of learning in computers and people*. Cambridge University Press, 1982.

[Schank and Abelson, 1977] R. Schank and R. Abelson. *Scripts, plans, goals and understanding*. Lawrence Erlbaum, 1977.

[Schank and Slade, 1991] R. C. Schank and S.C. Slade. The future of Artificial Intelligence: learning from experience. *Applied Artificial Intelligence* 5:97-107. Hemisphere Publishing Corporation, 1991.

[Segarra, 1995] J. Segarra. *Acquiring minimal concept descriptions and instance descriptions to discriminate between example sets*. Summer Project. Dept. of Computing Science. University of Aberdeen, 1995.

[Serra, 1993] P. Serra. *Development of a Knowledge-based System for control and supervision of urban wastewater treatment plants*. Ph.D. Thesis. Dept. de Química. Universitat Autònoma de Barcelona. (In Catalan), 1993.

[Serra *et al.*, 1995] P. Serra, M. Sànchez, J. Lafuente, U. Cortés and M. Poch. ISCWAP: A knowledge-based system for supervising activated sludge

[1]Accepted in January, 1996

processes. To appear in *Computers & Chemical Engineering,* Accepted in October, 1995.

[Serra *et al.*, 1994] P. Serra, M. Sànchez, J. Lafuente, U. Cortés and M. Poch. DEPUR: a knowledge based tool for wastewater treatment plants. *Engineering Applications of Artificial Intelligence* 7(1):23-30, 1994.

[Serra *et al.*, 1993a] P. Serra, J. Lafuente, R. Moreno, C. de Prada and M. Poch. Development of a real-time expert system for wastewater treatment plants control. *Control Eng. Practice* 1(2):329-335. Pergamon Press, 1993.

[Shinn, 1988] H.S. Shinn. Abstractional analogy: a model of analogical reasoning. *Proc. of Workshop on case-based reasoning (DARPA)*. Clearwater, Florida. 1988.

[Shirley, 1987] R.T. Shirley. Some Lessons learned using Expert Systems for Process Control. *IEEE Control Systems Magazine* 7(6):11-15, 1987.

[Sierra, 1989] C. Sierra. *MILORD: multi-level architecture for expert systems in classification*. Ph.D. Thesis. Dept. de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya, 1989.

[Simmons, 1990] R. Simmons. An achitecture for coordinating planning, sensing, and action. In *Proc. of the DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*. San Diego, CA, 1990.

[Simon, 1969] H.A. Simon. *The sciences of the artificial*. The MIT Press, 1969.

[Smets *et al.*, 1988] P. Smets, E Mamdani, D. Dubois and H. Prade. *Non-standard logics for automated reasoning*. Academic Press, 1988.

[Smith and Davis, 1981] R.G. Smith and R. Davis. Frameworks for cooperation in distributed problem solving. *IEEE Trans. on Systems, Man and Cybernetics* 11(1):61-70, 1981.

[Sriram, 1992] D. Sriram. Real time AI systems. In *Inteligencia Artificial y control en tiempo real*, Edited by Instituto de Ingeniería del Conocimiento–Repsol S.A., pp. 57-96, 1992.

[Sriram and Adey, 1987] D. Sriram and R.A. Adey. *Knowledge-based Expert Systems in Engineering: Tools and Techniques*. Computational Mechanics Publications, Ashurst, Southampton, U.K, 1987.

[Stankovic, 1988] J.A. Stankovic. Misconceptions about real-time computing: a serious problem for next generation systems. *IEEE Computer* 21(10):10-19, 1988.

[Steels, 1990] L. Steels. Components of expertise. *AI Magazine* 11(2):28-49, 1990.

[Steels, 1985] L Steels. Second Generation Expert Systems. *Future Generation Computer Systems*,1(4):213-221, 1985.

[Stefik *et al.*, 1982] M. Stefik, J. Aikins, R. Balzer, J. Benoit, L. Birnbaum, F. Hayes-Roth and E. Sacerdoti. The organization of expert systems: a tutorial. *Artificial Intelligence* 18:135-173, 1982.

[Stenstrom and Podunska, 1980] M.K. Stenstrom and R.A. Podunska. The effect of dissolved oxygen concentration on nitrification. *Water Reasearch* 14:643, 1980.

[Stephanopoulos, 1990] G. Stephanopoulos. Artificial Intelligence in Process Engineering: current State and future Trends. *Computers & Chemical Engineering* 14:1259-1270, 1990.

[Stephanopoulos and Stephanopoulos, 1986] G. Stephanopoulos and G. Stephanopoulos. Artificial Intelligence in the Development and Design of Biochemical Processes. *Trends in Biotechnology*, pp. 241-249, 1986.

[Stock, 1989] M. Stock. *Artificial Intelligence in Process Control*. McGraw-Hill, New York, 1989.

[Strom and Jenkins, 1984] P.F. Strom and D. Jenkins. Identification and significance of filamentous microorganisms in activated sludge. *Journal of Water Pollution Control Federation* 56:449-459, 1984.

[Sugeno, 1985] M. Sugeno. An Introductory Survey of Fuzzy Control. *Information Science* 36(1):59-83, 1985.

[Sun Microsystems, 1990] Sun Microsystems. *Sun Common Lisp 4.0 User's Guide*. September, 1990.

[Sussman, 1975] G.J. Sussman. *A computer model of skill acquisition*. American Elsevier, 1975.

[Sycara, 1987] K. Sycara. Finding creative solutions in adversarial impasses. *Proc. of 9th Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum, 1987.

[Tagaki, 1985] H. Tagaki. Fuzzy Identification of Systems and its Application to Modelling and Control. *IEEE Trans. on Systems, Man and Cybernetics* 15(1):116-132, 1985.

[Torra and Cortés, 1995] V. Torra and U. Cortés. Towards an automated consensus generator tool: EGAC. *IEEE Trans. on Systems, Man and Cybernetics* 25(5):888-894, 1995.

[Tzafestas and Ligeza, 1989] S. Tzafestas and A. Ligeza. A Framework for Knowledge Based Control. *Intelligent and Robotic Systems* 1(4):407-426, 1989.

[Valdés *et al.*, 1994] J. Valdés K. Peón, D. Hernández, A. García, R. Paredes, Y. García and A. Rodríguez. *The VEX-93 environment as a tool for developing* *knowledge systems with hybrid problem solving techniques*. Research Report. January, 1994.

[Van de Velde, 1987b] W. Van de Velde. *Learning from experience*. Ph.D. Thesis. Vrije Universiteit Brussel, 1987.

[Van de Velde, 1987a] W. Van de Velde. Inference structure as a basis for problem solving. In *Procc. of European Conference on Artificial Intelligence (ECAI-87)*, pp. 202-208, 1987.

[Van Harmelen, 1991] F. Van Harmelen. *Meta-level inference systems*. Pitman, 1991.

[Van Lehn, 1990] K. Van Lehn (editor). *Architectures for Intelligence*. Lawrence Erlbaum, 1990.

[Vedry, 1988] B. Vedry. *L'analyse biologique des boues activées*. Christian Brucker, SEGETEC, 1988.

[Veloso and Carbonell, 1993] M.M. Veloso and J.G. Carbonell. Derivational Analogy in PRODIGY: automating case acquisition, storage and utilization. *Machine Learning* 10(3):249-278, 1993.

[Venkatasubramanian, 1994] V. Venkatasubramanian. Towards Integrated Process Supervision: Current Status and Future Directions. In *Procc. of 2nd IFAC Workshop on Computer Software Structures Integrating AI/KBS Systems in Process Control*, pages 9-21, Lund, August, 1994.

[Vere, 1980] S. Vere. Multilevel counterfactuals for generalizations of relational concepts and productions. *Artificial Intelligence* 14:138-164, 1980.

[Voss, 1988] H. Voss. Architectural Issues for Expert Systems in Real Time Control. *IFAC Worksop on Artificial Intelligence in Real Time Control*, 1988.

[VUB AI Lab, 1993] VUB AI Lab. VUB Artificial Intelligence Laboratory, 10 years. Vrije Universiteit Brussel, 1993.

[Wanner *et al.*, 1987] J. Wanner, J. Chudoba, K. Kuckman and L. Proske. Control of activated sludge filamentous bulking: effect of anoxic conditions. *Water Research* 21:1447-1451, 1987.

[Weinberg *et al.*, 1992] J. B. Weinberg, G. Biswas and G. R. Koller. Conceptual clustering with systematic missing values. *Procc of 9th Int. Workshop on Machine Learning*, pp. 464-469, 1992.

[Wess and Globig, 1994] S. Wess and Globig. Case-based and symbolic classification - a case study. In *Topics in case-based reasoning* (S. Wess, K.D. Althoff and M. Richter, editors), LNAI-837, pp. 65-76, 1994.

[Wielinga *et al.*, 1992] B.J. Wielinga, A.T. Schreiber and J.A. Breuker. KADS: a modelling approach to knowledge engineering. *Knowledge Acquisition* 4(1):5-53, 1992.

[Winston, 1975] P.H. Winston. Learning structural descriptions from examples. In P.H. Winston editor, *The psychology of computer vision*. McGraw-Hill, New York, 1975.

[WPCF, 1990] WPCF. *Operation of Municipal Wastewater Treatment Plants. Manual of Practice*, 11, Water Pollution Control Federation, 2nd edition, 1990.

[WPCF, 1988] WPCF. *Aeration. Manual of Practice FD-13*, Nº 63, Chapter 6. Water Pollution Control Federation, Alexandria (Va), 1988.

[Wright, 1986] M.L. Wright. HEXSCON: A Hybrid Microcomputer-based Expert System for Real-Time Control Applications. *IEEE Western Conf. on Knowledge Engineering and Expert Systems*, pp. 49-54, 1986.

[Wright *et al.*, 1986] M.L. Wright, M.W. Green, G. Fiegl and P.F. Cross. An Expert System for Real-time Control. *IEEE Software* 3(2):16-24, 1986.

[Young, 1982] S.J. Young. Real Time Languages: Design and Development. Ellis Horwood, Chichester, 1982.

[Zadeh, 1983] L.A. Zadeh. The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and Systems* 11:199-277, 1983

[Zadeh, 1979] L.A. Zadeh. A theory of Approximate Reasoning. *Machine Intelligence* 9:149-194, 1979.