ONTOWEDSS - AN ONTOLOGY-BASED ENVIRONMENTAL DECISION-SUPPORT SYSTEM FOR THE MANAGEMENT OF WASTEWATER TREATMENT PLANTS

by

Luigi Ceccaroni

Supervisors: Ulises Cortés and Miquel Sànchez-Marrè

SUBMITTED TO OBTAIN THE DEGREE OF
DOCTOR IN ARTIFICIAL INTELLIGENCE
AT
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONA, SPAIN
OCTOBER 2001

To Iain and Piluco

Contents

List of Tables **ix**

List of Figures **xi**

1 Introduction **1**

1.1 Motivations 1

1.2 Thesis' structuring 5

I State of the art 7

2 Ontologies **9**

2.1 Philosophical background 11

2.2 AI definitions 12

2.3 Introducing the basic terminology 15

2.3.1 Programming conventions 16

2.4 Review of the field of ontologies 20

2.4.1 Design principles and microworlds 21

2.4.2 Pragmatics on knowledge sharing and reuse 22

2.4.3 Ontological commitment 24

2.4.4 Methodologies 24

2.4.5 Types 26

2.4.6 Languages used to codify ontologies 28

2.4.7 Applications and projects 34

2.4.8 Ontology editors 43

2.4.9 Problems, tradeoffs and solutions 45

2.4.10 Resources 46

3 Wastewater treatment process **49**

3.1 Economical aspects of AI in wastewater management 50

3.2 WWTP management 57

3.2.1 Wastewater treatment plants 57

3.2.2 Wastewater characterization 60

3.2.3 Aerobic biological waste-treatment 64

3.2.4 Suspended-growth systems 65

3.2.5 Micro-organisms in WWTPs 67

3.2.6 Process performance 69

3.2.7 Sampling, instrumentation and control 69

3.3 Classic control of the activated-sludge process 71

3.3.1 Classic control's problems 73

4 Environmental Decision Support Systems **77**

4.1 Rule-based expert systems 79

4.1.1 RBES and wastewater environment 83

4.1.2 RBESs' problems 83

4.2 Experiential knowledge and *case-based reasoning* (CBR) 85

4.2.1 CBR and wastewater environment 88

4.2.2 CBR's problems 89

4.3 Features of EDSSs 90

4.4 Examples of EDSSs 93

4.5 AI-paradigm integration 99

II An ontology-based environmental-DSS **103**

5 OntoWEDSS **105**

5.1 Knowledge acquisition 108

5.2 Architecture 109

5.3 Perception layer 110

5.3.1 Awareness 110

5.3.2 Descriptors and data processing 114

5.4 Diagnosis layer 118

5.4.1 Knowledge-based paradigm 120
5.4.2 Ontology 125
5.4.3 Diagnosis integration 126
5.5 Decision-support layer 129
5.6 Why OntoWEDSS is better than earlier systems? 131

6 WaWO **135**
6.1 Why an ontology for wastewater treatment? 135
6.2 Categories 137
6.3 Ontolingua vs. OIL: strategies for achieving a standard 146
6.4 WaWO's important features 149
6.5 Methodology overview 152
6.6 Formalization of WaWO in the Ontolingua language 154
6.7 Reuse of existing ontologies 157
6.8 Reasoning and problem-solving 158
6.9 Knowledge sharing 159

7 Implementation **163**
7.1 Perception layer 164
7.2 Diagnosis layer 171
7.2.1 RBES 171
7.2.2 CBRS 178
7.2.3 Ontology 183
7.3 Decision-support layer 196
7.4 How to use OntoWEDSS 198
7.4.1 Domain modelling 198
7.4.2 Execution 203
7.5 Discussion on contributions 204

8 Evaluation **207**
8.1 Evaluation design 209
8.2 Why the bulking-sludge state? 211
8.3 Evaluation of the CBRS 212
8.4 Evaluation of the RBES 213

8.5 Evaluation of the complete system . . . 214
8.6 Discussion . . . 223

9 Conclusions **225**
9.1 Future work and open research lines . . . 229

A WaWO's features **234**
A.1 Identifying features . . . 234
A.2 Descriptive features . . . 235
A.3 Functional features . . . 238

B Sample database **239**

List of Tables

3.1 Classification of the physical, chemical and biological descriptors of wastewater. 61

3.2 Orders of magnitude of response time for some fundamental parameters. 73

5.1 Descriptors of WWTPs as they appear in the WaWO ontology. . . . 112

5.2 Discretization of the descriptors of Granollers WWTP. 118

5.3 Classification of WWTP's operational states 119

6.1 Part of the Ontolingua library 153

8.1 CBRS's evaluation. 212

8.2 RBES's evaluation. 213

8.3 Evaluation of CBRS plus RBES. 213

8.4 Evaluation of CBRS plus RBES plus WaWO. 222

B.1 Database schema of Granollers WWTP with sample values. 239

List of Figures

2.1 Geographical categories 10
2.2 A view of an example ontology. 15

3.1 Scheme of a waste water treatment plant with activated sludge. 58
3.2 Aeration-tank processes. 66

4.1 A scheme of an expert system. 81
4.2 Knowledge acquisition in RBESs. 83
4.3 The CBR paradigm. 86
4.4 General scheme of an environmental decision-support system. 91

5.1 Main decision tree for filamentous-bulking problems (simplified). 107
5.2 Part of a decision tree. 108
5.3 Top-level decomposition of OntoWEDSS. 110
5.4 The OntoWEDSS architecture. 111
5.5 Fuzzy graph of a membership function. 123
5.6 Data flow of general diagnosis-integration. 127

6.1 Top-level categories in the WaWO ontology. 137

7.1 Window for the introduction of microbiological data through the API of the RBES of OntoWEDSS. 169
7.2 Window for the creation of rules in the RBES of OntoWEDSS. 174
7.3 Window for the creation of a domain in the CBRS of OntoWEDSS. 179
7.4 Comparison of CBRS and RBES results. 197
7.5 Window for the result of the rule-based expert system. 205

Abstract

The contributions of this thesis bridge two disciplines: environmental science (specifically, wastewater management) and computer science (specifically, artificial intelligence). Wastewater management as a discipline operates using a range of different approaches and methods which include: manual control, on-line automatic control, numerical or non-numerical models, statistical models and simulation models. The thesis characterizes an interdisciplinary research on artificial intelligence techniques (rule-based reasoning, case-based reasoning, ontologies and planning) applied to environmental decision-support systems. The integrated architecture's design of this application, the OntoWEDSS system, augments classic reasoning systems (rule-based reasoning and case-based reasoning) with a domain ontology about the management of wastewater treatment plants. The integration of the newly created WaWO ontology provides a more flexible management capability to OntoWEDSS. The construction of the OntoWEDSS decision support system is based on a specific case study but the system is also of general interest, given that its ontology-underpinned architecture can be applied to any wastewater treatment plant and, at an appropriate level of abstraction, to other environmental domains. The OntoWEDSS system improves the diagnosis of the state of a treatment plant, provides support for wastewater-related complex problem-solving, and facilitates knowledge modelling and reuse by means of the WaWO ontology.

The following research targets have been achieved in particular: (1) the improvement of the modelling of the information about wastewater treatment processes and the clarification of a part of the existing terminological confusion in the domain, (2) the incorporation of ontology-modelled microbiological knowledge related to the treatment process into the reasoning process, (3) the creation of a decision support

system with three layers (perception, diagnosis and decision support) which combines knowledge through a novel integration between KBSs and ontologies, providing better results, (4) the solution of existing reasoning-impasses, found using the new microbiological knowledge encoded in the hierarchical structure and the relations of the ontology, (5) the representation of cause-effect relations, due to the implementation of a set of relations that enable the ontology to automatically deduce the answer to questions about the wastewater domain.

OntoWEDSS is implemented in the LISP programming language, using Allegro Common LISP software. A focussed evaluation of the system, founded on the assessment of the capacity of response to specific problematic situations, has been carried out and has given fine results.

Resumen

Las contribuciones de esta tesis unen dos disciplinas: ciencias ambientales (específicamente, gestión de aguas residuales) e informática (específicamente, inteligencia artificial). El tratamiento de aguas residuales como disciplina opera utilizando un rango de diferentes enfoques y métodos que incluye: control automático on-line, modelado numérico o no-numérico, razonamiento basado en reglas, razonamiento basado en casos, soporte a la decisión y planificación. La tesis caracteriza una aplicación interdisciplinaria de técnicas de inteligencia artificial a sistemas de soporte a la decisión en el dominio ambiental. El diseño de la arquitectura de esta aplicación, el sistema OntoWEDSS, aumenta los sistemas híbridos de razonamiento ya existentes (razonamiento basado en reglas y basado en casos) con una ontología de dominio para la gestión de plantas de tratamiento de aguas residuales. La integración de la ontología WaWO, de nueva creación, proporciona a OntoWEDSS una mayor flexibilidad en la capacidad de gestión. La construcción del sistema de soporte a la decisión OntoWEDSS se basa en el estudio de un caso específico, pero el sistema resulta también es de interés general puesto que la arquitectura basada en ontologías puede aplicarse a cualquier planta de tratamiento de aguas residuales y, a un nivel apropiado de abstracción, a otros dominios ambientales. El sistema OntoWEDSS mejora la diagnosis del estado de la planta de tratamiento, proporciona soporte a la resolución de complejos problemas relacionados con aguas residuales, y facilita el modelado del conocimiento y su reutilización mediante la ontología WaWO.

En particular, la investigación ha alcanzado los siguientes objetivos: (1) la mejora del modelado de la información sobre procesos de tratamiento de aguas residuales y la clarificación de parte de la confusión existente en la terminología relacionada, (2) la incorporación de conocimiento microbiológico (referente al proceso del tratamiento y modelado mediante una ontología) dentro del proceso de razonamiento, (3) la

creación de un sistema de soporte a la decisión con tres estratos (percepción, diagnosis y soporte a la decisión) que combina conocimiento mediante una novedosa integración entre KBSs y ontologías, proporcionando mejores resultados, (4) la eliminación de obstáculos existentes en el razonamiento, hallada utilizando el nuevo conocimiento microbiológico codificado en la estructura jerárquica y las relaciones de la ontología, (5) la representación de relaciones causa-efecto, debido a la implementación de un conjunto de relaciones que permiten a la ontología deducir automáticamente la respuesta a cuestiones sobre el dominio de aguas residuales.

OntoWEDSS está implementada en el lenguaje de programación LISP, usando el software Allegro Common LISP. Se ha llevado a cabo una evaluación enfocada del sistema, basada en la valoración de la capacidad de respuesta a situaciones problemáticas específicas, obteniéndose buenos resultados.

Resum

Les contribucions d'aquesta tesi uneixen dues disciplines: ciències ambientals (específicament, gestió d'aigües residuals) i informàtica (específicament, intel·ligència artificial). El tractament d'aigües residuals com a disciplina opera fent servir un rang de diferents enfocaments i mètodes que inclouen: control manual, control automàtic on-line, modelat numèric o no-numèric, models estadístics i simulacions. La tesi caracteritza la recerca interdisciplinària de tècniques d'intel·ligència artificial (raonament basat en regles, raonament basat en casos, ontologies i planificació) a sistemes de suport a la decisió a l'entorn ambiental. El disseny de l'arquitectura d'aquesta aplicació, el sistema OntoWEDSS, augmenta els sistemes clàsics de raonament existents (raonament basat en regles i basat en casos) amb una ontologia de domini per a la gestió de plantes de tractament d'aigües residuals. La integració de l'ontologia WaWO recentment creada proporciona a OntoWEDSS una major flexibilitat en la capacitat de gestió. La construcció del sistema de suport a la decisió OntoWEDSS es basa en l'estudi d'un cas específic, però el sistema també és d'interès general ja que l'arquitectura basada en l'ontologia pot aplicar-se a qualsevol estació depuradora i, a un nivell apropiat d'abstracció, a altres dominis ambientals. El sistema OntoWEDSS millora la diagnosi de l'estat de l'estació depuradora, proporciona suport a la solució de complexes problemes relacionats amb aigües residuals, i facilita el modelatge del coneixement i la seva reutilització mitjançant l'ontologia WaWO.

En particular, a la investigació s'han aconseguit els següents objectius: (1) la millora del modelatge de la informació sobre processos de tractament d'aigües residuals i la clarificació de part de la confusió existent en la terminologia del domini, (2) la incorporació de coneixement microbiològic (referent al procés del tractament i modelat mitjançant una ontologia) dins del procés de raonament, (3) la creació

d'un sistema de suport a la decisió amb tres nivells (percepció, diagnosi i suport a la decisió) que combina coneixement mitjançant una nova integració entre KBSs i ontologies, proporcionant millors resultats, (4) la eliminació d'obstacles existents en el raonament, obtinguda utilitzant el nou coneixement microbiològic codificat a l'estructura jeràrquica i a les relacions de l'ontologia, (5) la representació de relacions causa-efecte, degut a la implementació d'un conjunt de relacions que permeten a l'ontologia deduir automàticament la resposta a qüestions sobre el domini d'aigües residuals.

OntoWEDSS està implementada en el llenguatge de programació LISP, fent servir el software Allegro Common LISP. S'ha dut a terme una avaluació focalitzada del sistema, basada en la valoració de la capacitat de resposta a situacions problemàtiques específiques, obtenint-se bons resultats.

Sommario

Questa tesi contribuisce alla intersezione di due discipline: le scienze ambientali (specificamente, la gestione delle acque di rifiuto) e la informatica (specificamente, la intelligenza artificiale). Nel trattamento delle acque di rifiuto come disciplina si utilizzano diversi metodi, che includono: controllo manuale, controllo automatico online, modelli numerici o non-numerici e simulazioni. La tesi caratterizza un'applicazione interdisciplinare di tecniche di intelligenza artificiale a sistemi di aiuto alla decisione in campo ambientale. L'architettura di questa applicazione, il sistema OntoWEDSS, amplia i sistemi di ragionamento ibrido esistenti (ragionamento basato su un sistema di regole, ragionamento basato sull'esperienza, aiuto alla decisione e pianificazione) con un'ontologia di dominio per la gestione di depuratori di acque di rifiuto. L'integrazione dell'ontologia WaWO, di nuova creazione, fornisce a OntoWEDSS una maggiore flessibilità nella sua capacità di gestione. La costruzione del sistema OntoWEDSS si basa sullo studio di un caso specifico, però il sistema risulta anche di interesse generale dato che l'architettura basata su un'ontologia può essere applicata a un qualsiasi depuratore e, considerando un adeguato livello d'astrazione, ad altri domini ambientali. Il sistema OntoWEDSS migliora la diagnosi dello stato del depuratore, fornisce aiuto alla soluzione di problemi complessi relazionati con le acque di rifiuto e facilita la modellizzazione della conoscenza e la sua riutilizzazione mediante l'ontologia WaWO.

In particolare, la ricerca realizzata ha raggiunto i seguenti obiettivi: (1) il miglioramento dell'informazione sui processi di depurazione e il chiarimento di parte della confusione esistente nella terminologia relativa, (2) l'incorporazione di conoscenza microbiologica (riguardo al processo di depurazione e mediante la modellizzazione ontologica) nel processo di ragionamento, (3) la creazione di un sistema di aiuto alla decisione con tre livelli (percezione, diagnosi e aiuto alla decisione) che combina la

informazione mediante un nuovo tipo d'integrazione tra classici sistemi basati sulla conoscenza e ontologie, proporzionando risultati migliori, (4) l'eliminazione di alcuni ostacoli esistenti nel ragionamento, ottenuta utilizzando la nuova conoscenza microbiologica codificata nella struttura gerarchica e nelle relazioni dell'ontologia, (5) la rappresentazione di relazioni causa-effetto del mondo reale attraverso l'implementazione di un insieme di relazioni ontologiche che permettono di dedurre automaticamente le risposte a domande sul dominio delle acque di rifiuto.

OntoWEDSS è implementata nel linguaggio di programmazione LISP, usando il software Allegro Common LISP. È stata realizzata una valutazione del sistema basata sulla stima della capacità di risposta a situazioni problematiche specifiche e si sono ottenuti risultati soddisfacenti.

The acronym world

AAAI	American national conference on AI
API	application program interface
AT	aeration tank
ATP	automated theorem prover
BOD	biochemical oxygen demand
CBR	case-based reasoning
CBRS	case-based reasoning system
CF	certainty factors
CLIM	common LISP interface manager
CLOS	common LISP object system
CORBA	common object request broker architecture
CycL	Cyc language
DAI	distributed AI
DAML	DARPA agent markup language
DARPA	defense advanced research projects agency
DCHEM	distributed chemical emergencies manager
DLP	description logic prover
DS	decision support
DSS	decision support system
DTD	document type definition
ECAI	European conference on AI
EDSS	environmental decision support systems
EWCBR	European workshop on case-based reasoning

ES	expert system
FaCT	fast classification of terminologies
FIPA	foundation for intelligent physical agents
F/M	food to micro-organism ratio
FRS	frame representation system
GA	genetic algorithms
GFP	generic frame protocol
GKB	generic knowledge-base
HC-REMA	health-care resource management
HPKB	high performance knowledge-base program
I	influent
IAAI	conference on innovative applications of AI
ICCBR	international conference on case-based reasoning
IJCAI	international joint conference on AI
ISI	Information Sciences Institute
IST	information society technologies programme
KB	knowledge base
KBMT	knowledge-based machine translation
KBS	knowledge-based systems
KBSI	Knowledge Based Systems, Inc.
KEML	knowledge-engineering and machine-learning group at UPC
KQML	knowledge query manipulation language
KRS	knowledge representation system
KSE	knowledge sharing effort
LISP	list processing language
LOOM	a description logic (ISI, University of Southern California)
LSI	llenguatges i sistemes informàtics
MCRT	mean cell-residence-time $\simeq$ SRT
MLSS	mixed liquor suspended solids
MLVSS	mixed liquor volatile suspended solids

OKBC	open knowledge base connectivity
ONIONS	ontological integration of naive sources
OntoWEDSS	ontology-based environmental decision-support system for wastewater
P	primary-treatment effluent
PATMAN	patient workflow management system
PBSA	planning balance sheet analysis
PLC	programmable logic controller
PTTP	Prolog technology theorem prover
RAS	recycled activated sludge
RBES	rule-based expert system
RBR	rule-based reasoning
RDF	resource description framework
RDFS	resource description framework schema
SCADA	supervisory control and data acquisition
SCBA	social cost benefit analysis
SRT	sludge residence time
SS	suspended-solids
SUO	standard upper ontology
SVI	sludge volumetric index
TAMBIS	transparent access to multiple biological information sources
TKN	total Kjeldahl nitrogen
TMR	text meaning representation
TOVE	Toronto virtual enterprise
TSS	total suspended-solids
UML	unified modelling language
UPC	Universitat Politècnica de Catalunya
WaRP	wastewater reactive planner
WAS	waste activated sludge
WATERSHEDSS	water, soil and hydro-environmental decision support system
WaWAT	wastewater agent town

WaWO	wastewater ontology
XHTML	extensible hypertext markup language
XML	extensible markup language
XMLS	extensible markup language schema
XOL	XML-based ontology-exchange language

Acknowledgments

My last year in college, I read Hofstadter's *Gödel, Escher, Bach* and I was caught by the artificial intelligence world. Now, after five years of thinking, talking and working in this field, I believe I achieved tangible results. I believe I can sketch an outline of a decision support system for wastewater, which integrates ontologies, rule-based reasoning and case-based reasoning. In these years of research I have had a lot of help from the people working and living with me. I wish to thank Diego, Leonardo, Glyn and Yannis for the feedback after having read a preliminary version of this thesis; Quim for helping in many occasions; Dave, Ignasi and Virginia for the stimulant analyses about the general idea of the thesis; Pilar, Mon, Camilo and Luis Carlos for their help with the figures and the constant support. The modesty of other people, who helped me with their invaluable advice, drove them to ask me not to be mentioned here. Regretting it, I fulfill their desire.

Chapter 1

Introduction

Al principio la Fe movía montañas sólo cuando era absolutamente necesario, con lo que el paisaje permanecía igual a sí mismo durante milenios.
Pero cuando la Fe comenzó a propagarse y a la gente le pareció divertida la idea de mover montañas, éstas no hacían sino cambiar de sitio, y cada vez era más difícil encontrarlas en el lugar en que uno las había dejado la noche anterior; cosa que por supuesto creaba más dificultades que las que resolvía.
La buena gente prefirió entonces abandonar la Fe y ahora las montañas permanecen por lo general en su sitio.
Cuando en la carretera se produce un derrumbe bajo el cual mueren varios viajeros, es que alguien, muy lejano o inmediato, tuvo un ligerísimo atisbo de Fe.

Augusto Monterroso

1.1 Motivations

This thesis presents the design and implementation of an *ontology*-based environmental decision-support system (named <u>OntoWEDSS</u>[1]) applied to the domain of wastewater treatment. This is an *innovative and interdisciplinary approach* to the management of knowledge in the problem-solving processes related to environmental

[1]Ontology-based Wastewater Environmental Decision-Support System

issues. In fact, even if the application studied is specific, the architecture presented could serve as a basis for any environmental system.

Throughout the thesis, we will touch many research areas: ontologies, rule-based expert systems, case-base reasoning, wastewater treatment, chemical and microbiological processes, and decision support systems.

With respect to *Artificial Intelligence* (hereafter, AI), the main contribution of the thesis is the study of the introduction and integration of an ontology with case-based reasoning and rule-based reasoning into an environmental decision-support system, together with the implementation of the system and an evaluation of the advantages related to the proposed approach. *Environmental Decision-Support Systems* (EDSSs) are useful when dealing with complex environmental problems, with processes which are not easily modelled because our knowledge is still incomplete and uncertain. The introduction of an ontological component in an EDSS lets us develop issues which will contribute to the improvement of the current state of the art in wastewater management:

- more stable wastewater treatment operation through an ontology-based supervision;
- *portability* of the management system of a *wastewater treatment plant* (hereafter, WWTP).

Ontologies have been developed in AI to facilitate knowledge sharing and reuse. They are a popular research topic in various communities, such as knowledge engineering, natural language processing, cooperative information systems, information integration, software agents, and knowledge management. In general, ontologies provide (Fensel *et al.*):

- a shared and common understanding of a domain; this domain can be communicated among people and across application systems;

- an explicit conceptualization (i.e., meta information) that describes the semantics of the data.

Recent articles covering the main aspects of ontologies in AI research are: Uschold and Gruninger (1996), van Heijst *et al.* (1997), Studer *et al.* (1998), Benjamins *et al.* (1999), Gómez-Pérez and Benjamins (1999a) and Fensel (2000).

In this thesis, we present the *WaWO ontology* (Ceccaroni *et al.* 2000a), which has been designed and built following the mainstream ideas about ontology construction (Uschold and Gruninger 1996; Uschold 1998a). WaWO is a hierarchically structured set of terms and a set of relations (see §2.3) describing the domain of wastewater treatment. WaWO is the manifestation of a shared understanding of the wastewater domain that is agreed among a number of experts in environmental and chemical engineering. The introduction of an agreed-upon ontology in the domain of wastewater biological treatment facilitates:

1. an accurate, effective *communication and sharing* of meanings, which in turn leads to other benefits such as knowledge reuse;

2. an advancement in the environmental technologies for the management of biological and biochemical processes;

3. enhancing the knowledge about the specific microbial ecology of environmental processes developing in the technological ecosystems of treatment plants.

Even though WaWO was designed on the basis of the specification of a few particular plants, the knowledge which it embodies is valid for any treatment plant of the same class. In the thesis, we will describe its structure in detail, discuss the approach it takes to some important issues in domain modelling and show how it handles a variety of example problems.

The WaWO ontology belongs to a more general decision support system for the supervision of WWTPs: *OntoWEDSS*. The OntoWEDSS system is a living[2] system which is part of the knowledge and technology needed for the rational management of water resources. OntoWEDSS receives on-line inputs[3] from on-line sensors all over the treatment plant as well as off-line inputs from the plant's laboratories and human operators. The system uses its internal knowledge-bases and inference mechanisms to process this information, to diagnose the ongoing state of the treatment plant and to predict the evolution of that state. Eventually, the output of the system is represented by statements about actions to be taken, or statements to support a human manager's decisions in future actuations, or direct control signals to treatment-plant's devices in order to maintain the plant operating correctly.

Wastewater treatment plants are the physical element of the domain modelled by the ontology and managed by OntoWEDSS. WWTPs serve to decontaminate wastewaters prior to their discharge into a natural body of water. For that, they use techniques of physical, chemical and biological treatment. The wastewater-treatment process is very *complex* and it is difficult to develop a reliable supervisory technology based only on a classic chemical-engineering control approach. As it will be explained in next chapters, the introduction of AI systems led to better results in wastewater management and process automation.

Rule-based expert systems, case-based reasoning systems and ontologies proved, individually, to be able to cope with some known difficulties and to face successfully several problems related to the wastewater domain. Larger knowledge bases could significantly improve current expert systems and tutoring systems. They should contain a *broad* knowledge of the domain, required to perform multiple tasks and to take into account multiple viewpoints in dealing with domain problems. And

[2]*Living* in the sense that it evolves.

[3]Input/Output devices are any of various devices used to enter information and instructions into the OntoWEDSS system for storage or processing and to deliver the processed data to a human operator or, in some cases, to a machine controlled by the system. Such devices comprise sensors and effectors. Apparatus of this kind with direct connection to OntoWEDSS's central processing unit is said to be on-line; peripheral equipment working independently of it is termed off-line.

another great improvement could come *combining* different modelling and reasoning systems.

The ontology developed in this thesis is a step forward in these directions. The adopted architecture (OntoWEDSS) integrates on-line and off-line data, and three knowledge-based systems[4] (KBSs) and it is flexible enough to deal with the complexity of the wastewater treatment process, given an adequate amount and type of data. In OntoWEDSS, the deep knowledge of the domain is represented and the evolution of micro-organism communities (a key element in biological treatment process) is taken into account. With OntoWEDSS it is possible to capture, understand and describe the knowledge about the whole physical, chemical and microbiological environment of a wastewater treatment plant.

In perspective, on one side the ontology represents a first step on the way for a *real portability* of the system towards other similar domains and it could be effectively employed to address the problem of general model-construction in domains close to the one of wastewater (*generalization*); while by the other side it is possible to instantiate/adapt the ontology to the specific configuration of a treatment plant and to automatically construct and validate specific models (*specification*).

1.2 Thesis' structuring

This thesis is organized as follows. Chapter 2 provides general background on ontologies. Chapter 3 describes the environmental domain: (a) the general wastewater treatment process and its possible variations and (b) the wastewater from a physical, chemical and biological point of view. The role of micro-organisms in the biological treatment of a wastewater treatment plant, the processes inside an aerobic biological

[4]Computer programs that use AI techniques to make decisions or recommendations or to predict outcomes based on an analysis of data. An example of knowledge-based system is an expert system, which typically has two parts: a very large database that contains specified knowledge in a given area and a set of rules for reaching conclusions. Often the rules are an elaborate set of *if-then* statements. Expert systems have been applied to environmental issues, chemistry, geology, genetic engineering, medicine, and pharmacology.

reactor and the activated-sludge environment are also discussed in this chapter. The conventional automatic-control system for the wastewater treatment and the parameters that it controls are also sketched. Chapter 4 is concerned with environmental decision-support systems. We compare rule-based expert systems and case-based reasoning systems: (a) their application to environmental issues, (b) their advantages and problems and (c) the dynamic learning of knowledge. We present the state-of-the-art in decision-support systems applied to environmental issues: what they do and which features they include. We explain how different AI paradigms integrate to implement a decision-support system and how to adopt an interdisciplinary approach to environmental problem solving. In Chapter 5 we explain how the decision support system (OntoWEDSS), in which the WaWO ontology is embedded, has been designed and we include a description of its layered architecture. In Chapter 6 we introduce the WaWO ontology. Its architecture and features are considered, as are the associated problems and the functioning of its components. Chapter 7 provides detail on the implementation of OntoWEDSS. Chapter 8 describes the evaluation of the system. Finally, the contributions of this work are discussed and ideas for future work are provided in Chapter 9. The appendix provides the feature taxonomy of the WaWO ontology that we constructed.

Part I

State of the art

Chapter 2

Ontologies

Era una vez una Cucaracha llamada Gregorio Samsa que soñaba que era una Cucaracha llamada Franz Kafka que soñaba que era un escritor que escribía acerca de un empleado llamado Gregorio Samsa que soñaba que era una Cucaracha.

Augusto Monterroso

In logic, the existential quantifier $\exists$ is a notation for asserting that something exists. But logic itself has no vocabulary for describing the things that exist. *Ontologies* fill that gap: they are used to study the existence of all kinds of entities, abstract and concrete, that make up the world (Sowa 2000).

In a more and more Web-based information society, high-level automatic data-processing requires a machine-understandable representation of information's semantics. This semantics is not provided by XML-based languages themselves (not to speak of HTML). Ontologies fill the gap again, providing sharable structure (see Figure 2.1) and semantics of a given domain, and therefore they play a key role in such research areas as knowledge management, electronic commerce, decision support and agent communication.

In the commercial and industrial domains, there are many companies which have

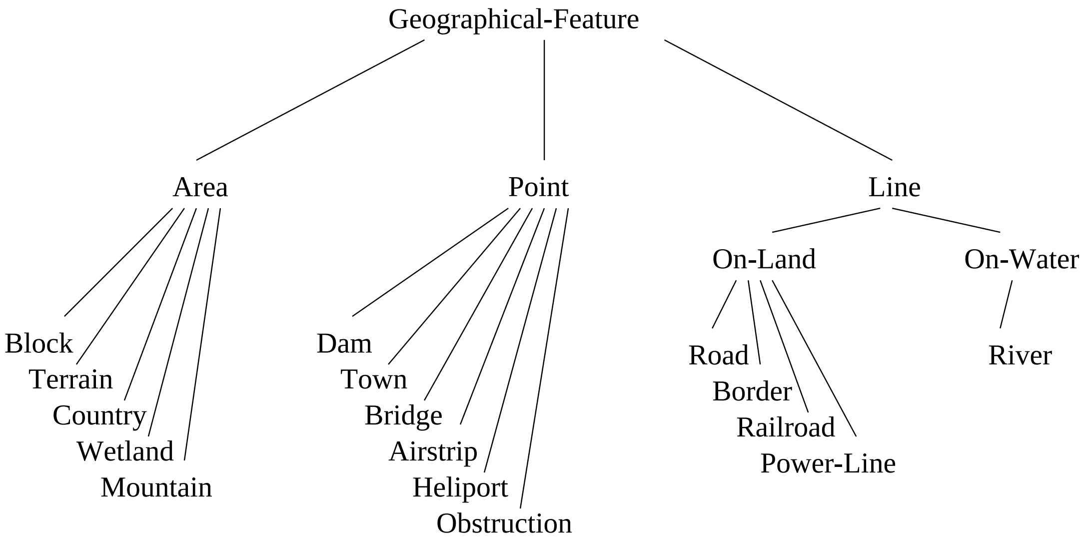

Figure 2.1: Geographical categories in the Chat-80 system (Warren and Pereira 1982).

a heavy legacy with the industrial era, whose culture prevents knowledge sharing. Nowadays, however, more and more companies share and manage knowledge as a strategy to create value. They are companies formed in the last decade, which try to work according to the principles of managing knowledge to add value to their businesses. In these companies, ontologies are used to formalize intellectual capitals and intangibles, and to represent and organize domain-knowledge after mergers. They are also used for data-source integration in global information systems and for in-house communication.

In recent years, there has been a considerable progress in developing the conceptual bases for building ontologies. They allow reuse and sharing of knowledge components, and are, in general, concerned with static domain-knowledge. Because of this, they are often used in association with *problem-solving methods* (PSMs), which deal instead with dynamic, reusable reasoning (Arpírez Vega *et al.* 2000). Ontologies and PSMs are basic means to enable reuse and sharing, of knowledge and reasoning behavior, across domains and tasks. Ontologies and PSMs can be used as complementary reusable components to construct knowledge-based systems (van Heijst *et al.* 1997). Ontologies provide a shared and common understanding of a domain, and PSMs describe the reasoning process of a knowledge-based system, in a domain and implementation independent fashion. In any case, many frameworks for building ontologies allow the inclusion of reasoning capabilities directly into the ontologies, with limited need of external PSMs.

2.1 Philosophical background

The origin of the term *ontology* can be found in philosophy:

- Ontology is the theory or study of being, i.e., of the basic characteristics of all reality. Though the term was first coined in the 17th century, ontology is synonymous with metaphysics or *first philosophy* as defined by Aristotle in the 4th century BC. Because metaphysics came to include other studies (e.g., philosophical cosmology and psychology), ontology has become the preferred term for the study of being. It was brought into prominence in the 18th century by Christian Wolff (1679-1750), a German rationalist, for whom it was a deductive discipline leading to necessary truths about the essences of beings. His great successor Immanuel Kant (1724-1804), however, presented influential refutations of ontology as a deductive system and of the ontological argument for God's necessary existence (as a supreme and perfect being). With the 20th-century renovation of metaphysics, ontology or ontological thought has again become important, notably among phenomenologists and existentialists, like Martin Heidegger (1889-1976).

- Ontology is a branch of the science of metaphysics which investigates and explains the nature and essential properties and relations of all beings, or the principles and causes of being. (from Webster's Revised Unabridged Dictionary)

- Ontology is the metaphysical study of the nature of being and existence. (from

WordNet)

To summarize, an *ontology* is a part of metaphysics: it is the science of the existence which investigates the structure of *being* in general, rather than analyzing the characteristics of particular beings. But what is *being*? As a test, Quine (1992) proposed his famous criterion: "To be is to be the value of a quantified variable". Again, as Quine's critics have noted, his criterion says nothing about what actually exists. Those who object to it would prefer some guidelines for the kinds of legal statements. For the purpose of this thesis, Quine's criterion can be used to determine the ontological commitment of WaWO representation. But, in general, further analysis is necessary to give the knowledge engineer some guidelines about what to say and how to say it. This is what we deal with in this chapter.

2.2 AI definitions

The AI literature is full of different definitions of the term *ontology.* Each community seems to adopt its own interpretation according to the use and purposes that the ontologies are intended to serve within that community. A catalog by Yannis Kalfoglou, available on-line[1], helps us to summarize the evolution of the definition of the term.

- One of the early definitions is: 'An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary.' (Neches *et al.* 1991)

- A widely used definition is: 'An ontology is an explicit specification of a conceptualization.' (Gruber 1993)

[1]http://www.dai.ed.ac.uk/homes/yannisk/seke99panelhtml.html

- An analysis of a number of interpretations of the word *ontology* (as an informal conceptual system, as a formal semantic account, as a specification of a conceptualization, as a representation of a conceptual system via a logical theory, as the vocabulary used by a logical theory and as a specification of a logical theory) and a clarification of the terminology used by several other authors is in Guarino and Giaretta (1995).

- An elaboration of Gruber's definition is: 'Ontologies are defined as a formal specification of a shared conceptualization.' (Borst *et al.* 1997)

- 'An ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base.' (Swartout *et al.* 1996)

- A paper, with an explanation of the terms used in early definitions, states: 'Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared refers to the notion that an ontology captures consensual knowledge, that is, it is not primitive to some individual, but accepted by a group.' (Studer *et al.* 1998)

- A working definition is: 'An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are interrelated which collectively impose a structure on the domain and constrain the possible interpretations of terms. An ontology is virtually always the manifestation of a shared understanding of a domain that is agreed between a number of agents. Such agreement facilitates accurate and effective communication of meaning, which in turn leads to other benefits such as inter-operability, reuse and sharing.' (Uschold 1998a)

- Recently, a broad definition has been given: 'We consider ontologies to be domain theories that specify a domain-specific vocabulary of entities, classes, properties, predicates, and functions, and to be a set of relationships that necessarily hold among those vocabulary terms. Ontologies provide a vocabulary for representing knowledge about a domain and for describing specific situations in a domain.' (Fikes and Farquhar 1999)

In this thesis, we adopt the following definition for *ontology*: **A formal and explicit specification of a shared conceptualization, which is readable by a computer**. It is derived from Gruber (1993), Borst (1997) and Studer *et al.* (1998). *Conceptualization* refers to an abstract model of some phenomenon in some world, obtained by the identification of the relevant concepts of that phenomenon. *Shared* reflects the fact that an ontology captures consensual knowledge and is accepted by a relevant part of the scientific community. *Formal* refers to the fact that an ontology is an abstract, theoretical organization of terms and relationships that is used as a tool for the analysis of the concepts of a domain. *Explicit* refers to the type of concepts used and the constraints on their use. Therefore, an ontology provides a set of well-founded constructs that can be leveraged to build meaningful higher level knowledge. The terms in an ontology are selected with great care, ensuring that the most basic (abstract) foundational concepts and distinctions are defined and specified. The terms chosen form a complete taxonomic set and the relationships among terms are defined using formal techniques. It is these formally defined relationships that provide the semantic basis for the terminology chosen. Although taxonomy contributes to the semantics of a term in a vocabulary, ontologies include richer relationships between terms. These rich relationships enable the expression of domain-specific knowledge, without the need to include domain-specific terms.

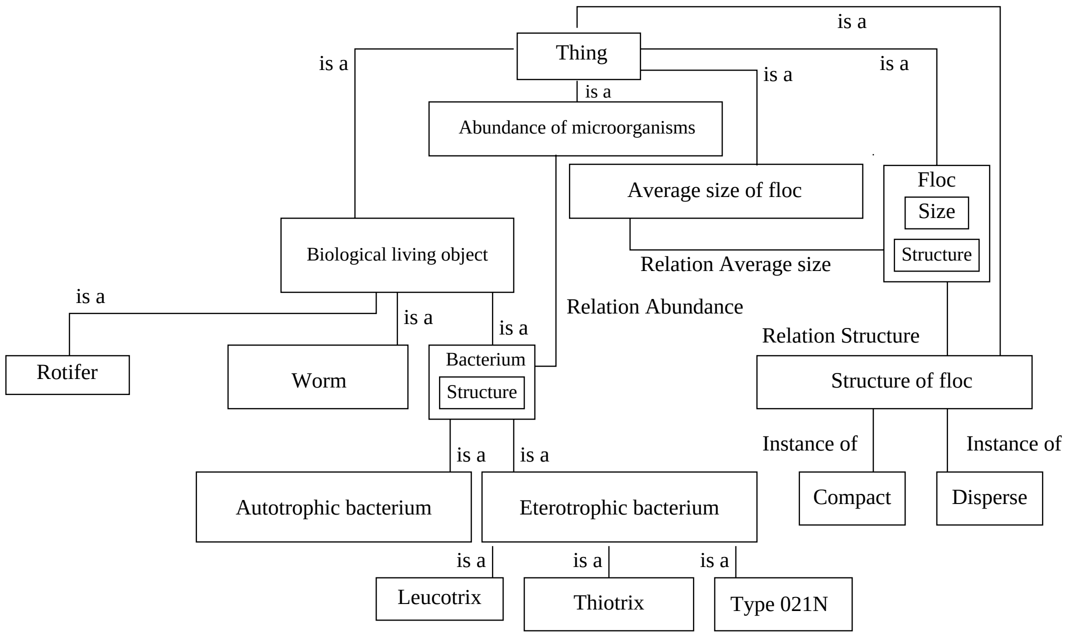

Figure 2.2: A view of an example ontology.

2.3 Introducing the basic terminology

Ontologies are good candidates for providing the shared and common domain structures which are required for semantic integration of information sources. Even if it is still difficult to find consensus among ontology developers and users, some agreement about protocols, languages and frameworks exists. In this section we clarify the terminology which we will use throughout the thesis. This meta-explanation is necessary, given that every researcher adopts and adapts whatever language and developing environment better suit his needs.

Ontologies (of which a basic example is shown in Figure 2.2) have a function similar to a *database* schema with which are often confused (see Klein *et al.* (2000) for an elaborated comparison between database schemas and ontologies). Some differences are (Fensel *et al.*):

- a language for defining ontologies is syntactically and semantically richer than common approaches for databases;

- an ontology must be a shared and consensual terminology because it is used for information sharing and exchange; a database does not have to;

- an ontology provides a domain theory, as opposed to a database, which provides the structure of a data container.

For a widespread use of ontologies in information integration and exchange, a prerequisite is the achievement of a joint standard for describing ontologies. Any approach that tries to achieve such a standard has to answer these questions (Fensel *et al.*): "What are the appropriate modelling primitives for representing ontologies?", "How can we define their semantics?" and "What is the appropriate syntax for representing ontologies?" A recognition of the importance of such issues is the setting up of the DAML program[2], which aims at the definition of a semantics of information sources with which machines can deal.

2.3.1 Programming conventions

In this section the basic terms and constructs which will be used later are introduced and explained. The reader will find that the following descriptions are sometimes biased by the framework that we will adopt for ontology construction (see §6.4).

- **Axioms** are the elements which permit the detailed modelling of the domain. There are two kinds of axioms that are important for this thesis: defining axioms and related axioms. *Defining axioms* are defined as relations in the *KIF-meta* Ontology[3]. A defining axiom for a constant (e.g., a symbol) is a sentence that helps defining the constant. *Related axioms* are defined as relations in the Frame Ontology[4]. A *Related-Axiom* is a multivalued relation

[2]DARPA Agent Markup Language (DAML) is a new Defense Advanced Research Projects Agency (DARPA) military (sob!) initiative, whose goal is to create technologies that will enable software agents to dynamically identify and understand information sources, and to provide interoperability between agents in a semantic manner. It formally commenced with a kick-off meeting on August 2000. Significant information will be published on http://daml.semanticweb.org.

[3]http://www.aiai.ed.ac.uk/~entprise/enterprise/ontology-code/kif-meta/

[4]http://java.stanford.edu/concur/examples/html-lib/frame-ontology/

(as opposed to a function) that maps any object in the domain of discourse to a KIF sentence (see §2.4.6) related to that object. An object is not necessarily a symbol. It is usually a class, or relation or instance of a class. If not otherwise specified, with the term *axiom* we refer to a related axiom.

- A **class** or **type** (e.g., *Rotifer* in Figure 2.2) is a set of objects. Each of the objects in a class is said to be an instance of the class. In some frameworks an object can be an instance of multiple classes. A class can be an instance of another class. A class which has instances that are themselves classes is called a *meta-class*. The top classes employed by a well developed ontology derive from the root class *object*, or *thing*, and they themselves are objects, or things. Each of them corresponds to the traditional concept of *being* or *entity*. A class, or *concept* in description logic (see §2.4.7), can be defined intensionally in terms of descriptions that specify the properties that objects must satisfy to belong to the class. These descriptions are expressed using a language that allows the construction of composite descriptions, including restrictions on the binary relationships connecting objects. A class can also be defined extensionally by enumerating its instances. Classes are the basis of knowledge representation in ontologies. Class hierarchies might be represented by a tree: branches represent classes and the leaves represent individuals.

- By **conceptualization** we mean a set of concepts, relations, objects and constraints that define the domain in question.

- An object-oriented **database schema** defines (1) a hierarchy of classes and (2) attributes and relationships of those classes.

- Objects that are not classes are referred to as **individuals** (e.g., *Compact* in Figure 2.2). Thus, the domain of discourse consists of individuals and classes, which are generically referred to as *objects*. Individuals are objects which cannot be divided without losing their structural and functional characteristics. They are grouped into classes and have slots. Even concepts like group or process can be individuals of some class.

- **Inheritance** through the class hierarchy means that the value of a slot for an individual or class can be inherited from its super class.

- A **knowledge base** is a collection of classes, individuals, slots, slot values, facets, and facet values. A knowledge base is also known as a *module*.

- Every class and every individual has a unique identifier, or **name**. The name may be a string or an integer and is not intended to be human readable.

- Following the assumption of anti-atomicity, **objects**, or *entities*, such as all nodes of Figure 2.2, are always *complex objects*. This assumption entails a number of important consequences. The only one concerning this thesis is that every object is a whole with parts (both as components and as functional parts). Additionally, because whatever exists in space-time has temporal and spatial extension, *processes* and objects are equivalent.

- **Relations** operate among the various objects populating an ontology. In fact, it could be said that the glue of any articulated ontology is provided by the network of dependency relations among its objects. The *class-membership* relation (called *instance-of*, see Figure 2.2) that holds between an instance and a class is a binary relation that maps objects to classes. The *type-of* relation is defined as the inverse of *instance-of* relation. If A is an *instance-of* B, then B is a *type-of* A. The *subclass-of* (or *is-a*) relation for classes is defined in terms of the relation *instance-of*, as follows: a class C is a subclass of class T if and only if all instances of C are also instances of T. The *superclass-of* relation is defined as the inverse of the *subclass-of* relation.

- Different users or any single user may define multiple ontologies within a single domain, representing different aspects of the domain or different tasks that might be carried out within it. Each of these ontologies is known as a **role**. In our approach we do not need to use roles, as we deal with only one ontology. Roles can be shared, as in Ontolingua shared sessions or as in Ontosaurus, or they can be represented separately in approaches without integration facilities. Roles can overlap in the sense that the same individuals can be classified in many different roles, but the class membership of an individual, its inherited slots and the values of those slots may vary from role to role. A representation of the similarities and differences between two or more roles is known as a *comparison*.

- Objects have associated with them a set of *own* **slots** (e.g., *Structure* in Figure 2.2) and each own slot of an object has associated with it a set of objects called *slot values*. Slots can hold many different kinds of values and can hold many at the same time. They are used to store information, such as name and description, which uniquely define a class or an individual. For example, the assertion that Iain's favorite food is `insalata di cavolo rosso` can be represented by the own slot `Favorite-Food` of the frame `Iain` having as value the frame `Insalata-Di-Cavolo-Rosso`. Classes have associated with them a collection of *template slots* that describe own slot values considered to hold for each instance of the class. The values of template slots are said to inherit to the subclasses and to the instances of a class. The values of a template slot are inherited to subclasses as values of the same template slot and to instances as values of the corresponding own slot. For example, the assertion that the gender of all female persons is female could be represented by the template slot `Gender` of class `Female-Person` having the value `Female`. If we create an instance of `Female-Person` called `Alicia`, then `Female` would be the value of the own slot `Gender` of `Alicia`. Own slots of an object have associated with them a set of *own* **facets**, and each own facet of a slot of a frame has associated with it a set of objects called *facet values*. For example, the assertion that the favorite food of Iain must be edible food can be represented by the facet `Value-Type` of the `Favorite-Food` slot of the `Iain` frame having the value `Edible-Food`. Template slots of a class have associated with them a collection of *template facets* that describe own facet values considered to hold for the corresponding own slot of each instance of the class. As with the values of template slots, the values of template facets are said to inherit to the subclasses and instances of a class. Thus, the values of a template facet are inherited to subclasses as values of the same template facet and to instances as values of the corresponding own facet.

- A **taxonomy** is a set of concepts, which are arranged hierarchically. A taxonomy does not define attributes of these concepts. It usually defines only the *is-a* relationship between the concepts. In addition to the basic *is-a* relation, the *part-of* relation may also be used.

- A **type** is an ontological category in AI (in which it is synonymous of *class*) and in logic.

- A **vocabulary** is a language dependent set of words with explanations/documentation. It seeks universality and formality in a local context (for example an environmental domain).

2.4 Review of the field of ontologies

Now that we have introduced the term *ontology* and the basic terminology of the field, we present a comprehensive review of this area of research. To this aim, we partially reuse the classification presented by Kalfoglou (2000), which we extend. In section §2.4.1 the design principles are explored; section §2.4.2 deals with knowledge sharing and reuse and section §2.4.3 with the ontological commitment; methodology issues are described in section §2.4.4 and the different types of ontologies are explored in section §2.4.5. Section §2.4.6 is about the languages used to codify ontologies; emphasis on applications is given in section §2.4.7; and ontology editors are explored in section §2.4.8. Finally, potential problems are discussed in section §2.4.9 and pointers to resources for further reading are given in section §2.4.10.

2.4.1 Design principles and microworlds

Philosophers usually build their ontologies from the top down. They start with grand conceptions about everything in heaven and earth (Sowa 2000). Programmers, however, tend to work from the bottom up. For their AI systems, they start with limited ontologies or *microworlds*, which have a small number of concepts that are tailored for a single application. The *blocks world*, with its ontology of blocks and pyramids, has been popular for prototypes in robotics, planning, machine vision and machine learning.

To design both a microworld or a top-level ontology, a number of criteria have been proposed and analyzed (e.g., by Gruber (1995)). They are outlined in the following list (Kalfoglou 2000):

1. *Clarity.* It refers to the effective communication of the intended meaning. Formalism has been proposed as a means to dissipate ambiguities. For example, when a definition can be stated as a logical axiom, it should be. However, all definitions should be documented in natural language.

2. *Coherence.* It means that the ontology should endorse all the inferences that are consistent with the axioms. Not only the defining axioms should be logically consistent, but the concepts that are defined informally (such as documentation and examples) also should. If a sentence that can be inferred

from the axioms contradicts a definition or example given informally, then the ontology is incoherent.

3. *Extendibility.* The ontology should be designed to anticipate the shared uses of its vocabulary. One should be able to define new terms for special uses based on the existing vocabulary, in a way that does not require the revision of the existing definitions.

4. *Minimal encoding bias.* An encoding bias arises when representation choices are made purely for the convenience of notation or implementation. Encoding bias should be minimized because the ontology can be shared by agents or systems using different representation schemes and different implementation languages.

5. *Minimal ontological commitment* (see also §2.4.3). An ontology should make as few claims as possible about the world being modelled, allowing the parties using the ontology freedom to specialize and instantiate it as needed.

It has to be noted that ontology designers can not always comply with the above criteria. A number of tradeoffs can be necessary (Gruber (1995) and §2.4.9) and ways of compromising between well designed ontologies and applicability have been investigated (Borst *et al.* 1997).

2.4.2 Pragmatics on knowledge sharing and reuse

The issue of ease of reuse is the focal point of study in many research projects. In AI, ontologies were born to help in knowledge reuse and sharing: *reuse* means building new applications by assembling components already built, while *sharing* occurs when different applications use the same resources. Reuse and sharing have the following advantage: they are cost, time and resources effective. However, when sharing knowledge it is possible to come across problems related to:

- the conceptualization method (Gómez-Pérez *et al.* 1996);
- the shared vocabulary (e.g., libraries of ontologies);
- the format to exchange knowledge (e.g., KIF);
- the specific communication protocol (e.g., KQML[5] external interface).

When reusing knowledge, the most common problems are:

- the search for knowledge components: dispersion of ontologies in space;
- the heterogeneity of knowledge-representation formalisms and implementation languages, depending on where ontologies are stored;
- the heterogeneity of lexicons (e.g., ontologies are often formalized at a different level of detail);
- the heterogeneity of semantics;
- synonyms and hidden assumptions;
- terminological differences (addressed by an integration of various AI paradigms, such as ontologies, natural language processing and machine learning, with cognitive science) (Gómez-Pérez 1998);
- the choice of a knowledge component that does not match properly the system needs;
- expensive usage (people, hardware, software, time);
- no common format for presenting relevant information about ontologies, to help users to decide which ontology suits best their purpose (Arpírez Vega *et al.* 2000).

[5]KQML = Knowledge Query Manipulation Language.

2.4.3 Ontological commitment

The ontological commitment refers to agreements on the use of the shared vocabulary by the agents committed to the ontology. An agent commits to an ontology when its observable actions are consistent with the definition in the ontology (Gruber 1995). Commitment to a common ontology is a guarantee of consistency but not of completeness, with respect to queries and assertions using the vocabulary defined in the ontology. According to Guarino (1998), an ontological commitment should be made explicit when applying the ontology, in order to facilitate its accessibility, maintainability and integrity. This will lead to an increase of transparency for the application software which is based on that ontology. We do not develop further this issue because it is peripheral to the thesis and we shift our attention to the methodologies used to assemble ontologies.

2.4.4 Methodologies

The construction of an ontology is a time-consuming and complex task. Nowadays, numerous ontologies are being developed and used in various research fields. There are also ontology servers that collect a number of ontologies. Even if it is widely recognized that constructing ontologies, or domain models, is an important step in the development of *knowledge-based systems* (KBSs), what is lacking is a consensus for a uniform approach in designing and maintaining these ontologies. However, there exists a small but growing number of methodologies that specifically address the issue of the development of ontologies (see also Jones *et al.* (1998)). In particular, in a comprehensive review of the field, Uschold and Gruninger (1996) report on two methodologies used in the context of the *Enterprise* ontology and the TOVE project.

TOVE and Enterprise modelling. The *TOronto Virtual Enterprise* (TOVE) is a deductive *enterprise model* (EM), an extension of a generic EM. An EM is a computational representation of the structure, activities, processes, information, resources, people, behavior, goals and constraints of a business, government or other enterprise. It can be both descriptive and definitional. The role of an enterprise model is to achieve model-driven enterprise design, analysis and operation. The TOVE group developed a methodological approach for the construction of an EM based on the definition of Gruninger and Fox (1994), Gruninger and Fox (1995) and Uschold and Gruninger (1996): motivating scenarios, informal competency questions, terminology specification, formal competency questions, axiom specification, completeness theorems. The TOVE approach is most interesting for the emphasis on formal ontology evaluation and the means of performing this evaluation is provided in the form of completeness theorems. Starting from another point of view, Uschold and King describe a frame methodology for ontology construction, based largely on the experience of developing the *Enterprise* ontology (Uschold *et al.* 1998). They identify four main steps: purpose and scope identification, formalization, formal evaluation, and documentation. In common with most recent KBS-development methodologies, the Enterprise approach distinguishes between informal and formal phases of ontology construction. In fact, the *Enterprise* ontology does not explicitly deploy a formal evaluation procedure; this was the main focus of the methodology used in the context of the TOVE project.

Given the basic work on construction and evaluation methodologies by Uschold, Gruninger and colleagues, others have focussed on the preliminary phases of construction. *Methontology*, on the other hand, provides support for the entire life-cycle of ontology development.

Methontology. It enables experts and ontology makers who are unfamiliar with implementation environments to build ontologies from scratch. Initially described in Gómez-Pérez *et al.* (1996) and then updated in Fernández *et al.* (1997), Methontology identify the following activities in the development of an ontology: specification, knowledge acquisition, conceptualization, integration, implementation, evaluation, documentation. The life cycle of the ontology is based on the refinement of a prototype and ends with a maintenance state. The most distinctive aspect of Methontology is the focus on this maintenance (Gómez-Pérez 1994; Gómez-Pérez 1995). The environment for building ontologies using the Methontology framework is called ODE (Ontology Design Environment). ODE is a software tool to specify ontologies at the knowledge level. ODE allows developers to specify their ontology by filling in tables and drawing graphs. It has a module which automatically translates the specification of the ontology into target languages.

An overview of methodologies used in AI projects along with a comparison with standards from system-engineering literature is given by Fernández (1999).

2.4.5 Types

The development methodologies reported above were used in some of the ontologies which will be described in section §2.4.7. Before proceeding to survey actual implementations of ontologies, we describe the various types in which ontologies are classified in the literature.

Categorization can be done in terms of *generality*.

- For example, ontologies like UpperCyc (Lenat and Guha 1990) or SUO[6] model generic notions that form the foundations for knowledge representation across various domains. These are called *top-level ontologies*, like Sowa's one (Sowa 2000).

- On the other side there are the *domain ontologies*: small-scale, specific ontologies which are carefully tailored to a specific domain. Examples of this type are the *PhysSys* ontology (Borst *et al.* 1997), which captures knowledge regarding physical system processes, and the WaWO ontology described in this

[6] http://suo.ieee.org/

thesis, used to represent wastewater-treatment knowledge.

Another classification of ontologies is concerned with their *purpose*.

- There exist *task* ontologies that capture task-related knowledge independently of the domain in which a task is defined.

- Complementary to these ones are the *method* ontologies which provide definitions of the relevant concepts and relations used to specify a reasoning process to achieve a particular task.

- Of yet another type are the *knowledge representation* ontologies. The most representative example is the Frame ontology (Gruber 1993) which captures the representation primitives used in frame-based languages and allows other ontologies to be specified using frame-based conventions.

- Another type includes the *linguistic* ontologies. The most illustrative examples are the Generalized Upper Model (GUM) (Bateman *et al.* 1995), WordNet (Miller 1990) and SENSUS (Knight and Luk 1994). This sort of ontologies are often called *terminological* ontologies, whereas ontologies like TOVE are called *axiomatized* ontologies.

With respect to the degree of *formality*, in their overview of the field, Uschold and Gruninger (1996) identified the following types: highly informal, semi-informal, semi-formal, rigorously formal. In the *informal classes* there are more or less structured definitions in natural language. In the *formal classes* there are ontologies defined through artificial formal languages (e.g., *Ontolingua*) or first order theories with formal semantics, theorems and proofs of such properties as soundness and completeness (e.g., TOVE).

2.4.6 Languages used to codify ontologies

Any language used to codify ontology-underpinned knowledge should be expressive, declarative, portable, domain independent and semantically well defined. Here, we present the most representative examples.

GFP (Karp *et al.* 1995)

- The *Generic Frame Protocol* (GFP), jointly developed at SRI International and Knowledge Systems Laboratory of Stanford University, provides a set of functions that support a generic interface to underlie *frame representation systems* (FRSs). The interface layer allows an application to be independent from the idiosyncrasies of specific FRS software and enables the development of generic tools that operate on many FRSs.
- http://www.ai.sri.com/~gfp/

OKBC (Chaudhri *et al.* 1997; Chaudhri *et al.* 1998)

- GFP recently evolved into *open knowledge base connectivity* (OKBC) (developed under the sponsorship of the *High Performance Knowledge Base program*[7]), an API (Application Program Interface) for accessing ontologies stored in different *knowledge representation systems* (KRSs), not only systems that can be viewed as frame representations. OKBC provides a uniform model of KRSs based on a common conceptualization of classes, individuals, slots, facets and inheritance. OKBC is defined in a programming language independent fashion, and has existing implementations in Common LISP, Java, and C. For frame-based systems, OKBC is already a standard to transfer knowledge from one context to another. Its knowledge model supports features most commonly found in frame-based knowledge representation systems, object databases and

[7] HPKB

relational databases. OKBC has also been chosen by FIPA[8] as an exchange standard for ontologies.

- http://www.ai.sri.com/~okbc/

OKBC-Lite

- The OKBC-Lite knowledge model is an abridged version of the knowledge model used for the OKBC. OKBC-Lite extracts most of the essential features of OKBC, while not including some of its more complex aspects. OKBC-Lite plays a central role in XOL by defining the semantic framework in which XOL ontologies are defined. The ontology building blocks defined by the OKBC-Lite knowledge model include classes, individuals, slots, facets, and knowledge bases. The knowledge model also recognizes some basic data types. Although many aspects of the OKBC-Lite knowledge model are derived directly from the OKBC model, several simplifications to the OKBC model have been made. All references to the notion of *frame* have been eliminated because it does not have a clear portable definition. Only a subset of facets that are in most common use have been included.

- http://www.oasis-open.org/cover/xol.html
  http://www.ontoknowledge.org/oil/TR/existingwork.html

XOL

- XOL is a frame-based language (its modelling primitives and semantics are based on OKBC-Lite) with an XML syntax for the exchange of ontologies, proposed by the BioOntology Core Group. It is designed to provide a format for exchanging ontology definitions among a set of agents. The ontology definitions that XOL encodes include both schema information (meta-data),

[8] http://www.fipa.org

such as class definitions, and non-schema information (ground facts), such as object definitions. XOL is similar to other ontology-exchange languages and its development was inspired by Ontolingua. XOL differs from Ontolingua in having an XML-based syntax rather than a LISP-based syntax; the semantics of OKBC-Lite are similar to the semantics of Ontolingua.

- http://www.ai.sri.com/pkarp/xol/
  http://smi-web.stanford.edu/projects/bio-ontology/
  http://www.ontologos.org/Ontology/XOL.htm

CycL (Lenat and Guha 1990)

- CycL, the Cyc representation language, is a large and flexible knowledge representation language. It is essentially an extension of first-order predicate calculus, with extensions to handle equality, default reasoning, skolemization, and some second-order features. For example, quantification over predicates is allowed in some circumstances, and complete assertions can appear as intensional components of other assertions. CycL uses a form of circumscription, includes the unique names assumption, and can make use of the closed world assumption where appropriate.
- http://www.cyc.com/

KIF (Genesereth 1991)

- The *Knowledge Interchange Format* (KIF) is a language designed to be used in the interchange of knowledge among disparate computer systems (created by different programmers, at different times, in different languages). KIF is not intended as a primary language for interaction with human users (though it can be used for this purpose). KIF is based on predicate logic but provides a LISP-oriented syntax for it. Semantically, there are four categories of constants in KIF: objects, functions, relations and logical constants. Logical constants express conditions about the world and are either true or false. KIF is unusual among logical languages in the sense that there is no syntactic distinction among these four types of constants; any constant can be used anywhere. This feature allows the reification of formulas as terms used in other formulas, making it possible to make statements over statements. This introduces second-order features in KIF, which provides an important extension of first-order logic. Among the logic-based formats for exchanging knowledge, KIF, developed at Stanford University, is a standard in communication among ontologies.

- http://logic.stanford.edu/kif/kif.html

LOOM (MacGregor 1991)

- LOOM is a language and environment for building intelligent applications. The heart of LOOM is a knowledge representation system that is used to provide deductive support for the declarative portion of the LOOM language. Declarative knowledge in LOOM consists of definitions, rules, facts, and default rules. A deductive engine called a classifier utilizes forward-chaining, semantic unification and object-oriented truth maintenance technologies in order to compile the declarative knowledge into a network designed to efficiently support on-line deductive query processing.
- http://www.isi.edu/isd/LOOM/LOOM-HOME.html

Ontolingua language (Gruber 1993; Farquhar *et al.* 1997)

- The original Ontolingua language, as described by Gruber (1993), was designed to support the design and specification of ontologies with a clear logical semantics. To accomplish this, Gruber started from KIF and extended it with additional syntax, to capture intuitive bundling of axioms into definitional forms with ontological significance, and a Frame Ontology to define object-oriented and frame-language terms. The Ontolingua Server has extended the original language in two ways. First, it provides explicit support for building ontological modules that can be assembled, extended, and refined in a new ontology. Second, it makes an explicit separation between an ontology's presentation (the manner in which KIF axioms are viewed and manipulated by a user) and semantics (the underlying meaning).

The original Ontolingua language provided limited support for defining ontological modules in the form of a tree of named ontologies. Users found this simple model to be inadequate in several ways. Furthermore, the module system did not have a clearly defined semantics; this was in sharp conflict with the basic goals of the language.

The separation of presentation and semantics has always been implicit in Ontolingua's translation approach to sharing ontologies. In the current system, however, the explicit recognition of this distinction has become a key notion. The semantics of an ontology is always defined by a set of KIF axioms. In Ontolingua, the semantics is always simple, clear, and unambiguous. The presentation, in the Ontolingua Server's browsing and editing environment, is tailored for object-oriented or frame-language descriptions of the world. Farquhar *et al.* (1997) guarantee that each statement corresponds unambiguously to a KIF axiom. The vocabulary used in the presentation is defined in the Frame Ontology. The Frame Ontology defines terms including class, subclass-of, slot, slot-value-type, slot-cardinality, facet and so on. If an ontology is defined using this vocabulary, the Ontolingua Server can present it in a user-friendly form.

A key property of the extended Ontolingua Language and its presentation in the Ontolingua Server is that axioms that do not fit into the frame language are allowed. There is no restriction on expressiveness. This is extremely important for an ontology development environment. In contrast with an inference tool or a traditional knowledge representation tool for which tractability is paramount, an ontology development tool must support expressiveness.

The Ontolingua Server, however, must operate on ontologies and translate them into less expressive languages, in some cases. For this reason the editing environment encourages users to stay within the relatively simple frame sub-language. Commands for creating subclasses, adding slots, constraining slot values and so on, are easy to find and use. The frame-language axioms are presented simply and concisely. It is possible, however, for users to write arbitrary KIF axioms. Even if an axiom is untranslatable, it will still serve as an important formal specification of the authors intention. Indeed, because KIF and consequently Ontolingua is monotonic, performing translations into less expressive languages will still retain their correctness.

To summarize, the underlying representation for an ontology is a set of KIF axioms. These sentences are projected through a variety of lenses to produce the editor's frame pages, HTML documents, LOOM knowledge bases, Prolog clauses, and objects that can be manipulated using the GFP. The resulting Ontolingua language is a computer-interpretable description language which enables easy on-line collaborative construction of ontologies.

- http://ontolingua.stanford.edu/

2.4.7 Applications and projects

We now selectively make mention of some efforts of ontology construction and deployment, highlighting their contribution to the field. A complete listing of ontologies is impossible; the literature references are huge. Pointers to various resources are provided in section §2.4.10.

Ontolingua. Researchers of the Stanford University Knowledge Sharing Effort (KSE) Project developed Ontolingua (at the Knowledge System Laboratory), a tool to build ontologies on-line in a portable form, and worked on criteria to verify and evaluate knowledge sharing technology (Gómez-Pérez 1994; Gómez-Pérez 1995). In section §2.4.6 we dealt with the Ontolingua language; now we focus the attention on the Ontolingua ontology server. In 1999, the Stanford University Ontolingua Ontology Server (Farquhar *et al.* 1997) was a widely used tool for ontology construction and sharing[9], as it provides an extensive library of sharable ontologies whose definitions can be reused for developing of new ontologies. In Ontolingua, the class hierarchy is determined by the membership of individuals and classes to other classes, rather than being directly defined intensionally by the user. The manuals for the use of the Ontolingua Ontology Server (Farquhar *et al.* 1995; Farquhar *et al.* 1997; Fikes *et al.* 1997) contain advice on developing, browsing, maintaining and sharing ontologies on the Server. Ontolingua uses the Frame Ontology as a meta-ontology[10]. One of the main benefits in using the Ontolingua server is that it provides access to a library of previously defined ontologies. The library grows as developers add new ontologies to the repository.

Ontology construction in Ontolingua is based on the principle of modular development. Ontologies from the library can be re-used in four different ways:

[9] Today, the main other proposal for an ontology interchange language is OIL (see a comparison in § 6.3 on page 146)

[10] The *Related-Axioms* relation is used by Ontolingua translators to denote axioms related to a class, relation or instance that cannot be formulated using the Frame Ontology.

1. Inclusion: *ontology A* is explicitly included in *ontology B*. The vocabulary of *ontology A* is translated into the vocabulary of *ontology B*. This translation is applied to the axioms of *ontology A* too, and the translated axioms are added to *ontology B* (Farquhar *et al.* 1997). Multiple inclusion is supported.

2. Polymorphic refinement: a definition from an ontology is included in another ontology and refined. For example, the *Biological-Living-Object* class, defined in UpperCyc ontology as a subclass of *Composite-Tangible-And-Intangible-Object*, can be included in the WaWO ontology, renamed *Wastewater-Biological-Living-Object* and extended to be a subclass of *Wastewater-Microbiological-Taxonomy-Subdomain*.

3. Restriction: a restricted (by axioms) version of one ontology is included in another.

4. Cyclic inclusion: situations such as the following are allowed, although not recommended: *ontology A* is included in *ontology B*, *ontology B* is included in *ontology C* and *ontology C* is included in *ontology A*.

These distinctions are very useful in the re-use of ontologies. In 2001, Ontolingua is still one of the most used means of implementing ontologies, although a more comprehensive methodology needs to be used in conjunction with the Server.

OIL. The Ontology Inference Layer[11] (Fensel *et al.*) is a proposal for a joint standard for specifying and exchanging ontologies on the Web. OIL is entirely Web-driven and is based on:

[11]http://www.ontoknowledge.org/oil/

1. *Description logic* (DL)[12], which provides formal, clean and well defined *semantics* and efficient reasoning support. DL is part of a research effort in knowledge representation to provide theories and systems for expressing structured knowledge and for accessing and reasoning with it in a principled way.

2. Frame-based systems, which provide epistemologically[13] rich *modelling primitives.* OIL incorporates the essential modelling primitives of XOL (see §2.4.6) into its language. OIL is based on the notion of *concept* and the definition of its super-classes and attributes. Relations can also be defined not as attributes of a class, but as independent entities having a certain domain and range. Like classes, relations can be arranged in a hierarchy. While in DL roles are not defined for concepts (actually, concepts are defined as subclasses of role restriction), in a frame context a class is a subclass of its attribute definitions (i.e., all instances of the class must fulfill the restrictions defined for the attributes). Asking which roles could be applied to a class does not make much sense in DL, as nearly all slots can be applied to a class, while with frame-based modelling the implicit assumption made is that only those attributes which are defined for a class can be applied to that class. The ontology definitions encoded by XOL include both schema information, such as class definitions, and non-schema information, such as object definitions.

3. Existing standards such as OKBC (see §2.4.6), and new Internet standards such as XML[14] and RDF[15] which provide *syntactically* exchangeable notations.

[12]DL, also known as *terminological logics*, describes knowledge in terms of concepts and role restrictions that are used to automatically derive classification taxonomies. DL is the foundation of an important and powerful class of logic-based knowledge-representation languages (e.g., CLASSIC, CRACK, DLP, FaCT, GRAIL, KRIS, LOOM, PowerLOOM, RACE; see http://www.ida.liu.se/labs/iislab/people/patla/DL/systems.html for links to most papers and projects in the area of DL).

[13]*Epistemology* is the study of the origin, nature, and limits of human knowledge.

[14]XML (eXtensible Markup Language) is the universal format for structured documents and data on the Internet. The base specifications are XML 1.0 (W3C Recommendation, February 1998) and Namespaces (January 1999).

[15]RDF (Resource Description Framework) (http://www.w3c.org/Metadata/) is a foundation for

OIL is intended to improve OKBC, XML and RDF with necessary features for expressing rich ontologies and its core language has been designed so that it provides most of the modelling primitives commonly used in frame-based ontologies and automated reasoning support (e.g., class consistency and subsumption checking). OIL shares many *features* with OKBC and defines a clear semantics and XML-oriented syntax for them (extending OKBC). In the same way as OIL provides an extension of OKBC (and is therefore downwards compatible with it), OIL provides an extension of XML and RDF. Techniques for performance evaluation, developed for XML, can directly be used for ontologies specified in OIL because the XML syntax of OIL is defined by using the XMLS[16] mechanism. XMLS incorporates the notion of inheritance and this allows to capture the semantics of the *is-a* relationship. RDF and RDFS[17] are further candidates for a Web-based syntax for OIL. The relationship between

processing meta-data; it provides interoperability between applications that exchange machine-understandable information on the Internet. RDF uses XML to exchange descriptions of Internet resources but the resources being described can be of any type, including XML and non-XML resources. RDF emphasizes facilities to enable automated processing of Internet resources. RDF can be used in a variety of application areas, for example: in resource discovery to provide better search engine capabilities; in cataloging for describing the content and content relationships available at a particular web-site or digital library; by intelligent software agents to facilitate knowledge sharing and exchange; in content rating; in describing collections of pages that represent a single logical *document*; for describing intellectual property rights of web-sites; and for expressing the privacy preferences of a user as well as the privacy policies of a web-site. RDF provides the means for adding semantics to a document without making any assumptions about the structure of the document. RDF is an infrastructure that enables the encoding, exchange and reuse of structured meta data.

[16]XMLS (eXtensible Markup Language Schema) is a successor of DTD (Document Type Definition) and is published as a proposal by the W3C (http://www.w3.org/TR/1999/WD-xmlschema-2-19991217/). XMLS lets define the different pieces of data to model, along with the relationships among them. The ability to include this semantic information is the source of XML's power and its main advantage over HTML. XMLS definitions are themselves XML documents. XMLS provides a rich set of data-types that can be used to define the values of elementary tags. XMLS provides rich means for defining nested tags (i.e., tags with sub-tags). XMLS provides the *Namespaces* mechanism to combine XML documents with heterogeneous vocabulary. XMLS is still a proposal, it may change in the near future and currently does not provide much tool support.

[17]RDFS (Resource Description Framework Schema) provides a basic type schema for RDF. *Objects*, *classes*, and *properties* can be described. Predefined properties can be used to model *instance of* and *subclass of* relationships as well as domain restrictions and range restrictions of attributes. (D. Brickley and R.V. Guha: Resource Description Framework Schema Specification 1.0, W3C Candidate Recommendation 27 March 2000. http://www.w3.org/TR/2000/CR-rdf-schema-20000327)

OIL and RDFS is much closer than that between OIL and XMLS. This is not surprising, since XMLS was meant to generalize the way of defining the structure of valid XML documents and RDFS was meant to capture meaning in the way semantic nets do. In the same way as RDFS is used to define itself it can also be used to define other ontology languages. Therefore, a syntax for OIL is defined by giving an RDFS for the core of OIL and an extension to this RDFS is proposed to complement this core by covering further aspects.

Even if OIL is based on DL, verification and validation in OIL/XML are basically syntax-based, whereas in logic they are typically based on theorems. It is not yet clear how the OIL/XML/RDF community will deal with soundness and completeness. Nonetheless, for reasons we will present in §6.3, OIL proposers believe that the existing Ontolingua design for an ontology interchange language is not appropriate as a standard ontology language for the Internet and put forward OIL as an alternative and better standard.

Cyc. The Cyc Knowledge Server is a very large, multi-contextual ontology and inference engine developed by Cycorp (Guha and Lenat 1990; Lenat and Guha 1990). Cycorp's goal is to break the *software brittleness bottleneck* once and for all by constructing a foundation of basic *common sense* knowledge (a semantic substratum of terms, rules, and relations) that will enable a variety of knowledge-intensive products and services. Cyc is intended to provide a *deep* layer of understanding that can be used by other programs. The Cyc product family comprises also a set of interface tools and a number of special-purpose application modules running on Unix and Windows NT. The knowledge base is built upon a core of over 1,000,000 hand-entered assertions (or *rules*) designed to capture a large portion of what we normally consider consensus knowledge about the world. For example, Cyc knows that trees are usually outdoors, that once people die they stop buying things and that glasses of liquid should be carried rightside-up. This foundation enables Cyc to understand and reason about its application domains, for example: (1) Cyc can find the match between a user's query for *pictures of strong, adventurous people* and an image whose caption reads simply *a man climbing a cliff*, (2) Cyc can notice if an annual salary and an hourly salary are inadvertently being added together in a spreadsheet, (3) When someone searches for *Bolivia* on the Web, Cyc knows not to offer a follow-up question like *Where can I get free Bolivia online?*.

OpenCyc will be the open source version of the Cyc technology. Cycorp has set up an independent organization, OpenCyc.org[18], to disseminate and administer OpenCyc, and have committed to a pipeline through which all current and future Cyc technology will flow into ResearchCyc (a substantially larger subset of the Cyc Knowledge Base available for R&D in academia and industry) and then OpenCyc. Release 1.0 of OpenCyc will be delivered in 2002.

IDEF5. The IDEF5 method (http://www.idef.com/), developed by KBSI (Knowledge Based Systems, Inc.), is designed to assist in the general creation, modification and maintenance of ontologies. The IDEF5 ontology development process consists of the following five activities:

1. Organizing and scoping: this activity establishes the purpose, viewpoint and context for the ontology development project, and assigns roles to the team members.

2. Data collection: during this activity, raw data needed for ontology development are acquired.

3. Data analysis: it involves analyzing the data to facilitate ontology extraction.

4. Initial ontology development: this activity develops a preliminary ontology from the data gathered.

[18]http://www.opencyc.org/

5. Ontology refinement and evaluation: these activities complete the development process.

Supporting the ontology development process are *IDEF5's ontology languages*. There are two such languages: the IDEF5 schematic language and the IDEF5 elaboration language. The schematic language is a graphical language, specifically tailored to enable domain experts to express the most common forms of ontological information. This enables both to input the basic information needed for a first-cut ontology and to augment or revise existing ontologies with new information. The other language is the IDEF5 elaboration language, a structured language that allows detailed characterization of the elements in the ontology.

Reference Ontology and (ONTO)^{2}Agent. The creation of a class of *yellow pages* of ontologies is believed (Arpírez Vega *et al.* 2000) to be a solution to speed up the use of ontologies in applications. These living *yellow pages* provide classified and possibly up-to-date information about available ontologies and help users to locate candidate ontologies for a given application. (ONTO)^{2}Agent is an Internet broker specialized in the ontology field, which uses an ontology (the Reference Ontology) as its knowledge source, disseminates information about existing ontologies, helps to search appropriate ontologies and supplies pointers for the set of ontologies that meet user's requirements. The technology used to build the ontology-based Internet broker is called OntoAgent architecture and (ONTO)^{2}Agent is an instantiation of this architecture that answers questions in the domain of ontologies. These questions are about the features of the ontologies that have been entered into the Reference Ontology. A possible query would be: 'Give me all the ontologies in the domain *D* that are implemented in languages *L1* or *L2*'.

ONIONS. The ONIONS (ONtological Integration Of Naive Sources) methodology (Gangemi *et al.* 1996) is motivated by the knowledge integration problem, i.e., how to integrate heterogeneous sources of information in knowledge acquisition. One of the most distinctive aspects of the ONIONS approach is the production of a preliminary non-formal ontology, a schematic account of the conceptualization of a domain. Rather than focusing on the issue of a final representation of an ontology, ONIONS focuses on problems in knowledge acquisition and ontology refinement.

CommonKADS and KACTUS. CommonKADS (Schreiber *et al.* 1999) is a widely used methodology for the development of KBSs in which ontologies play an important role. The KACTUS project is a follow-up project which focuses on the issue of ontology development. An engineering approach is adopted, stressing modular design, redesign and reuse (Schreiber *et al.* 1995). New ontologies are constructed from a library of other small-scale ontologies, and this requires mapping among the various ontologies included in the development of the new ontologies.

Biology Ontology. The KBS group of the Department of Computer Sciences at the University of Texas at Austin has built a large ontology in the area of biology and developed methods for automatically answering a variety of questions using the correspondent ontology-underpinned KB. Containing about 30,000 concepts, this ontology is one of the largest of its kind (i.e., with structured and formally represented content). It is used for a variety of AI tasks, for example to test a system for explanation generation (Lester 1994).

TAMBIS Ontology. This ontology of biological terminology provides a model of biological concepts that can be used to form a semantic framework for many tasks of data storage, retrieval and analysis, in the bioinformatics domain. Such tasks can be the querying of heterogeneous sources or the systematic annotation of experimental results. Design and organization are considered (Baker *et al.* 1999) crucial for maintaining the coherence of the large collection of concepts of TAMBIS

(Transparent Access to Multiple Biological Information Sources) Ontology and their relationships. TAMBIS uses DL to represent knowledge and it is argued that DL is flexible and powerful enough to capture and classify concepts in a consistent way, and that DL can be applied to construct ontologies which can be used for making inferences from biological data.

No ontology application exists yet in the field of **wastewater** or in related fields and no ontology modelling the evolution of microbiological systems has been defined. We think that the representational power of ontologies can be exploited to deepen our knowledge about the micro-organisms of treatment plant's activated-sludge and the wastewater domain in general, and can be integrated together with other reasoning methods to improve the whole supervision of wastewater treatment (see §5.4).

2.4.8 Ontology editors

There are a number of more or less generic editors to create and manage ontologies. We provide here a few examples:

- The Stanford *Ontolingua Ontology Editor* (Stanford KSL Network Services[19]) was, in 1999, the most standard editor to create ontologies. It is a Web-based tool for creating, editing and browsing ontologies in the Ontolingua language (see §2.4.6).

- *OilEd* is a simple ontology editor developed by Sean Bechhofer at the University of Manchester. OilEd allows the user to: (1) build ontologies; (2) use the FaCT reasoner to check the consistency of ontologies and add implicit *subClassOf* relations; (3) export ontologies in a number of formats including both OIL-RDF and DAML-RDF. For further details and information about

[19]http://www-ksl-svc.stanford.edu

OIL, consult 2.4.7. The intention behind OilEd is to provide a simple, freeware editor that demonstrates the use of, and stimulates interest in, OIL. OilEd is not intended as a full ontology development environment. It does not actively support the development of large-scale ontologies, the migration and integration of ontologies and many other activities that are involved in ontology construction. Rather, offers just enough functionality to allow users to build ontologies and to demonstrate how the FaCT reasoner can be used to check and enrich ontologies. OilEd is available as freeware and is not fully supported or maintained. It is possible to download the installer for OilEd from http://img.cs.man.ac.uk/oil/. To get the full benefit from OilEd, it is also necessary to have the CORBA-FaCT reasoner installed[20]. The latest Windows version also includes the FaCT reasoner.

- *WebOnto*[21] (HC-REMA, PATMAN and Enrich projects) is an on-line tool for collaborative construction of ontologies.

- *Protégé-2000*[22] (Noy *et al.* 2000) is a tool for ontology editing and knowledge acquisition. Protégé-2000 has hundreds of users who use it for projects ranging from modelling cancer-protocol guidelines to modelling nuclear-power stations. Protégé-2000 is aimed at making it easier for knowledge engineers and domain experts to perform knowledge-management tasks. One of the major advantages of the Protégé-2000 architecture is that the system is constructed in an open source, modular fashion. Its component-based architecture enables system builders to add new functionality to Protégé-2000 by creating appropriate plug-ins such as support for alternative storage formats and domain-specific user-interface components. From Protégé, it is possible to export ontologies to other knowledge-representation systems, such as RDF, OIL and DAML.

[20] http://www.cs.man.ac.uk/FaCT/
[21] http://webonto.open.ac.uk/
[22] http://protege.stanford.edu/

- The *Generic Knowledge-Base* (GKB) *Editor*[23] is a graphical KB editor, implemented in LISP + CLIM (an interface-development module), from SRI, which supports the GFP. The GKB-Editor is a tool for editing and graphically browsing ontology underpinned KBs across multiple Frame Representation Systems in a uniform manner.

- *Ontosaurus Web Browser*[24] is an on-line tool for collaborative construction of ontologies. It is part of the Ontosaurus ontology-server (Swartout *et al.* 1996).

- *OntoEdit*[25] (Staab and Maedche 2000) is an off-line tool which enables developing, inspecting and modifying ontologies. It uses a GUI to codify conceptual structures (concepts, concept hierarchy, relations, axioms). Ontologies in OIL format can be imported and it is possible to export ontologies in OIL and F(rame)-Logic (Kifer *et al.* 1995) formats.

2.4.9 Problems, tradeoffs and solutions

Despite the fact that ontologies have been applied with success in a variety of fields, problems have been reported and attempts have been made to find solutions and tradeoffs. O'Leary (1997) raises the issues of formality in ontology development and argues for the difficulty in establishing a consensus among all the agents involved. Uschold *et al.* (1998) identify the problem of lack of translators when the representation formalisms used are not the same. They argue that the translation effort can be intensive and the lack of automatic support is an important disadvantage.

Ontologies are often designed to represent a microworld. The principal advantage of working with a specialized domain is ease of analysis, design and implementation. Its weakness, however, is the difficulty of sharing and reusing data and programs in other applications (Sowa 2000). Limited ontologies will always be useful for single

[23]http://www.ai.sri.com/~gkb/
[24]http://mozart.isi.edu:8003/sensus/sensus_frame.html
[25]http://www.ontoprise.de

applications in highly specialized domains. But, to share knowledge with other applications, an ontology has to be embedded within a more general framework. Philosophy provides that framework: its guidelines and top-level categories form the superstructure that can relate the details of the lower-level projects. If this framework is not implemented, the numerous problems may force users to stop reusing knowledge components and to formalize the same knowledge again. Due to these complications, relatively few applications in areas like knowledge management, ontology-based brokers, natural language generation, enterprise modelling, KBSs and inter-operability among systems reuse ontologies. At present, to ease the search for knowledge components to reuse and share, intelligent agents on the Internet are needed (Arpírez Vega *et al.* 2000). The need for this kind of services has been acknowledged and a web-site exists which gathers information about ontologies that have been built with the same logical organization, together with a broker specialized in the ontology field that helps in this search (Arpírez Vega *et al.* 2000).

Another important drawback is the lack of rigorous evaluation techniques for ontologies, and the problems related to maintenance have also been acknowledged by many researchers. Robertson (1998) summarizes that the cost of producing an ontology is not just in developing the domain-specific formal language, but also in maintaining it once the system is deployed. Over-commitment to defining specific details of an ontology can cause failure either after deployment or during sharing.

However, there are ways to alleviate the situation and solve some of the problems mentioned above. For example, with respect to shared-vocabularies problems, the online libraries of ontologies (e.g., *Ontolingua*) are a potential solution. Moreover, new frameworks have been proposed (Uschold 1998b; Uschold and Jasper 1999) to share experiences, discuss tradeoffs and disseminate knowledge regarding attempts to apply ontologies.

2.4.10 Resources

In this section we include pointers to publicly available online resources. They are all collections of ontology-related research projects.

- http://www.cs.utexas.edu/users/mfkb/related.html
- http://www.cs.man.ac.uk/∼franconi/ontology.html
- http://ksl-web.stanford.edu/kst/ontology-sources.html
- http://www.dai.ed.ac.uk/homes/yannisk/seke99panelhtml.html
- http://www.lsi.upc.es/∼luigic/ON-TO/ON-TO.htm
- http://www.kr.org/top/projects.html
- http://saussure.irmkant.rm.cnr.it/onto/link.html

In addition to these periodically updated online resources, there are several overviews in the literature. An early one with emphasis to applications of ontologies is in Uschold and Gruninger (1996). A comparative review of top-level ontologies with respect to design principles is in Fridman-Noy and Hafner (1997). The role of formal ontologies in information systems is reviewed in Guarino (1998). A recent review and survey of ontology research is in Chandrasekaran *et al.* (1999). A review of ontologies and PSMs is in Gómez-Pérez and Benjamins (1999b). A paper with emphasis to the role of ontologies in IT (Information Technology) is the one by Guarino and Poli (1995). A paper on ontologies and KBSs is the one by van Heijst *et al.* (1997). An editorial introduction to a specialized issue on the use of ontologies is the one by Uschold and Tate (1998).

In this chapter, we surveyed the main studies on ontologies and established the background for the chapters that follow. As a preview of what will be presented *in detail in chapter 6, we anticipate that the WaWO ontology follows in its first formulation the design principles of Ontolingua and has been created with the on-line Ontolingua Ontology Editor, taking into account all the available compatible guidelines on methodology coming from the ontological community.*

Chapter 3

Wastewater treatment process

> No one is ever as shocked and surprised ... as when the inevitable occurs.
>
> *Paul Baran*

The wastewater treatment process is part of the water cycle and, as such, it has a direct relation with other water systems or reservoirs. *Wastewater treatment plants* (WWTPs) receive water from the anthropic system of sewers, they somehow process it, and finally they deliver this water to a natural reservoir. The wastewater processing is what we care about, but we cannot forget the two other closest components of the global water cycle just mentioned (sewers, and rivers or sea).

It is on the basis of the quantity and quality of water to be treated that WWTPs are built, taking into account the possible fluctuations in the inflow[1]. These fluctuations can be very important where the sewerage system is not very developed and therefore it is not able to damp down inflow peaks towards the plant.

The main objectives in wastewater-treatment research are:

- knowing better the relevant characteristics of the wastewater;
- refraining the contaminated water from reaching the natural environment.

[1]Terminological clarification: in this thesis, with the term *inflow* we refer also to the term *influent* used in the literature

The fact is that continuously increasing economic and cultural pressures on freshwater resources, including pollution and excessive use, are causing threats which are augmenting costs and multiplying conflicts among different users of this strategic resource. These pressures can also impair the natural regenerative functions of the ecosystems in the water cycle. Two of the main challenges in the area of general water-management are to protect the water bodies and to provide high quality water in sufficient quantity at affordable costs. In order to achieve these goals, multidisciplinary research-efforts and actions are necessary. The very existence of WWTPs and the research for improving them goes in this direction and constitutes an essential element for an integrated sustainable management of water resources. The objectives of such sustainable management are to develop technologies to prevent and treat pollution of water, to purify water, to use and re-use it rationally, to enhance efficient treatment of wastewater and to minimize environmental impacts from wastewater treatment (including the prevention of potential health hazards).

3.1 Economical aspects of AI in wastewater management

There are several socio-economical aspects to be taken into account when dealing with the introduction of AI in wastewater management. Wastewater treatment involves different resources (water, soil, energy) and other aspects such as health. This calls for institutional co-operation on new levels either within or between institutions. The required integrated approach to resource management is still totally unfamiliar, both politically and socially, and is rarely practised by societal representatives.

Keeping the environment clean generates costs, which society and the private sector in particular tend to ignore for as long as possible. The environmental issue has only recently led to the implementation of relevant legislation. Effective law enforcement should be ensured. Few of the newly industrializing countries, where the problems are often the most severe, approach the protection of the natural resources.

It is even more difficult to regulate regional and global environmental problems (such as pollution of lakes and rivers) which require well respected international regulations.

The present predominant practice to handle waste and wastewater in ways hazardous to the environment and public health, as found in many countries, is not only the outcome of changes in production conditions, increases in population and urbanization, inadequate administrative and legal structures, but also relates to a lack of affordable technical solutions and alternatives. The concern about the crisis in water supply, problems of soil degradation and energy shortages has intensified the discussion about the merits of the different AI applications among wastewater-treatment technologies.

Wastewater-treatment technologies are used world-wide in different sectors with great variations in their objectives and applications. The opportunities offered by AI techniques are not equally known in all countries, whilst in some cases they raise unrealistic expectations.

The full potential of these techniques in terms of economic benefits (cost reductions), ecology (protection of water and soil) and social factors (health, availability of drinking water) has so far not been properly exploited. Some reasons for this are primarily technical (slow development of the technology), economic (exploitation of nature as a source of raw materials and a *rubbish bin*), institutional (lack of co-operation between sectors, insufficient know-how) and political.

Nevertheless, some of the preconditions favoring a wider use of AI techniques have changed in the course of the nineties. In chapter 4 we provide an overview of the current state and the development of these techniques. Here, we outline the reasons for their slow spreading despite significant ecological and economic advantages, and describes some non-technical factors that limit the application of AI techniques.

Lack of different applications for AI techniques. During the last two decades, there has been an increasing interest and technical improvement in the use of AI techniques for the treatment of sewage. But AI techniques for the treatment of liquid industrial effluents and household waste, as well as various mixed systems (e.g., co-fermentation[2]), are still virtually unavailable.

Economy and environment. The necessary capital to carry out the required investments is often still lacking and the small number of plants using AI decision-support systems implies that economies of scale cannot yet be achieved. Since public contractors often dominate the municipal sector, the most cost-effective solution is not always the one favored by decision-makers. The speed of decision-making in public institutions can barely keep pace with current technological and market developments.

AI techniques are relatively unknown and usually not taken into account in conventional wastewater-treatment management. The know-how required for planning, operation and adaptation is therefore rarely adequate. Professional organizations are often unaware of the potentials of these technologies, which causes a delay to their further development and appraisal.

Current status of wastewater treatment. In recent times, an increasing concern for the environment and a recognition of the economic advantages has brought about a boost in the use of state-of-the-art technology, at least in some industrialized countries (e.g., EU, Japan and USA[3]), in several larger Latin-American countries (e.g., Mexico), in part of Asia, as well as in Africa. In any case we are far away

[2]Fermentation of liquid manure with industrial sludge, municipal wastes or vegetable matter
[3]Before the Bush administration began the war against Afghanistan.

from reaching market saturation. Small- and medium-sized applications on a local level have so far been installed only in selected cases in few countries and are rarely developed and utilized in a systematic manner.

Recently, application of AI techniques, such as decision support systems, for municipal sewage treatment in towns and cities of different sizes is increasingly attracting the interest of national and international private institutions and is gradually being considered as an attractive and proven alternative to conventional control. In the context of improving cost-benefit relations and increased environmental awareness, AI techniques now receive considerably more attention. The implementation, however, is still hindered by a number of factors.

Application of AI techniques and know-how reuse. Certain elements of the technology, such as ontologies, are relatively new. Without the usual development, adjustment and refinement phases, suitable solutions have not yet evolved for all special needs.

Relatively few standardized schemes are so far available for small and medium-sized applications using AI techniques for the treatment of municipal and industrial wastewater. Sometimes, ideological and emotional arguments, rigidity based on outdated know-how and the slow pace of change among the responsible authorities, stand in the way of implementation of AI techniques. Appropriate *south-south* co-operation is, with few exceptions in Latin America, not well developed. As a result, knowledge transfer from countries where AI techniques are already established to other countries is not optimal.

Favorable factors. The introduction and increasing use of AI techniques in wastewater treatment and other domains can be attributed to various factors: the increased degree of sophistication and operational reliability of AI applications and software components; the heightening pressures to achieve savings in investment and operational costs; the growing know-how in this sector; the growing understanding of the interdependence of environmental resources and the necessity for institutional co-operation; the continuously expanding environmental and health problems; population growth and urbanization shortage of resources, leading to an increased interest in water treatment and recycling concepts.

Opportunities for action. Factors which inhibit the best possible and proper application of AI techniques can be categorized as follows and call for varying responses accordingly:

- Unclearness about the applicability of AI techniques. This factor is bound to continue as a hindrance to the application of AI as long as the knowledge about the potentials of the process is insufficient and the technological and economic aspects are not well evaluated.

- Subjective opposition (partly caused by prejudices, lack of information and ideological stereotyping), drawing on emotion, is very hard to deal with and to influence since it is rarely expressed directly and the line of argument constantly changes. It is only partially open to systematic counter-argument through information. Possible ways of dealing with this include: provision of clear and objective factual information, admission of the actual weaknesses and limitations of AI techniques, avoidance and removal of failures, well-presented documentation of successful projects, visits to plants, softening up the confrontational positions and making the discussion more objective by the presentation of the diverse facets of problems and their possible solutions.

- Objective private or commercial interests which oppose potentially more proper applications. These interests are equally difficult to deal with and to discuss and, in the case of institutions and companies, are usually of a financial nature, frequently based on experience in training courses, existing know-how, past experience, licenses and patents. They can to some extent be dealt with as follows: establishment of clear rules and regulations for the choice of alternatives in the tendering process (ensuring objectivity in the decision process), systematic provision of information to decision-makers and authorities which invite tenders, stronger co-operation and sharing of the information flow relating to technology, improvement in personnel training at traditional wastewater management companies and at the institutions/authorities inviting tenders.

- Current limitations of the technology itself and of its proper and more intensive utilization. These limitations can be modified through the implementation of technological, social, economic, information- related and planning measures.

  Current technological limitations are the outcome of a failure to adjust to local conditions, experience and know-how, as well as the technology's short span of experience and development. This can be rectified by: the establishment and documentation of suitable examples of working plants; the further development of the technology in terms of standardization and cost-reduction measures; practical research and development in the areas of post-treatment, pathogens removal, emission and odor control, gas utilization and sludge storage. *Social limitations* can be approached by: considering early separation of water, nutrients and harmful substances; setting and supervision of appropriate wastewater discharge standards; imposing regulations, which require an integrated approach to the management of environmental resources (soil, water, energy); support and documentation of inter- and intra-institutional co-operation; consideration of sustainability criteria; provision of documentation to support decision-making and submission of checkable criteria for decisions. *Economical limitations* can be dealt with through: standardization of plants; development of well adapted and standardized small- and medium-sized systems, which are commercially of little interest to the private sector; provision of information to key institutions involved in political and financial decision-making; securing appropriate funding lines and guarantees; increasing the numbers of plants in which *Environmental Decision-Support Systems* (EDSSs) are used, in order to achieve scale-effects; promotion of objective discussion concerning advantages and disadvantages of reusing system components.

Information and know-how availability can be supported by: establishing a network of specialists and institutions and providing appropriate information material through this network; detailed analysis of model plants and examples of integrated decision-support systems; broadening of the know-how base by organizing and supporting training courses in the area of plant maintenance; documentation and accessibility of training; provision of information to the (specialist) public.

Ways to improve the decision-making process, during the stage of planning which management system will be used, are: consideration of social and employment factors; inclusion of general reuse criteria in the choice of the technology.

The optimization of plants and decision support systems, information reuse, documentation and networking, as well as schooling and training, are likely to be the focus of attention in the near future, setting the scene for a more objective discussion and improved choice of technology in the interests of an improved wastewater and waste management.

3.2 WWTP management

3.2.1 Wastewater treatment plants

In a wastewater treatment plant (WWTP), the main goal is to reduce the level of pollution of the inflow water, that is to remove, within certain limits (depending on local legislation), too high amounts of pollutants in the water prior to its discharge to the natural environment. Nowadays the most widespread class of WWTP is a plant with physical-chemical treatment and an additional biological reactor (for better organic matter removal), which can be of two main sub-type, depending on the sort of growth of micro-organisms (Beccari 1991):

- *suspended growth*: with the micro-organisms mixed with the wastewater and dispersed in the form of free cells or of bio-flocks (activated sludge reactors);
- *attached growth*: with the micro-organisms anchored, in the form of bio-film, to inert surfaces (biological-film reactors).

The work of the thesis focuses on WWTPs with activated sludge (see Figure §3.1), which is now the most common case in the European Union. Such wastewater treatment process schematically consists of:

- Preliminary treatment (pre-treatment): the wastewater flows into the WWTP and undergoes a *screening* process (e.g., grit removal of voluminous solids and mechanical removal of floatable solids), chemical additions, (pre)aeration, *removal of sand* and foam, and *degreasing*. Odor control and flow measurement are carried out, too.
- Primary treatment: in treatment plants which receive highly contaminated waters, a physical treatment is carried out. In a settler, suspended solids form flocs, sediment on the bottom of the tank in the form of sludge and are taken

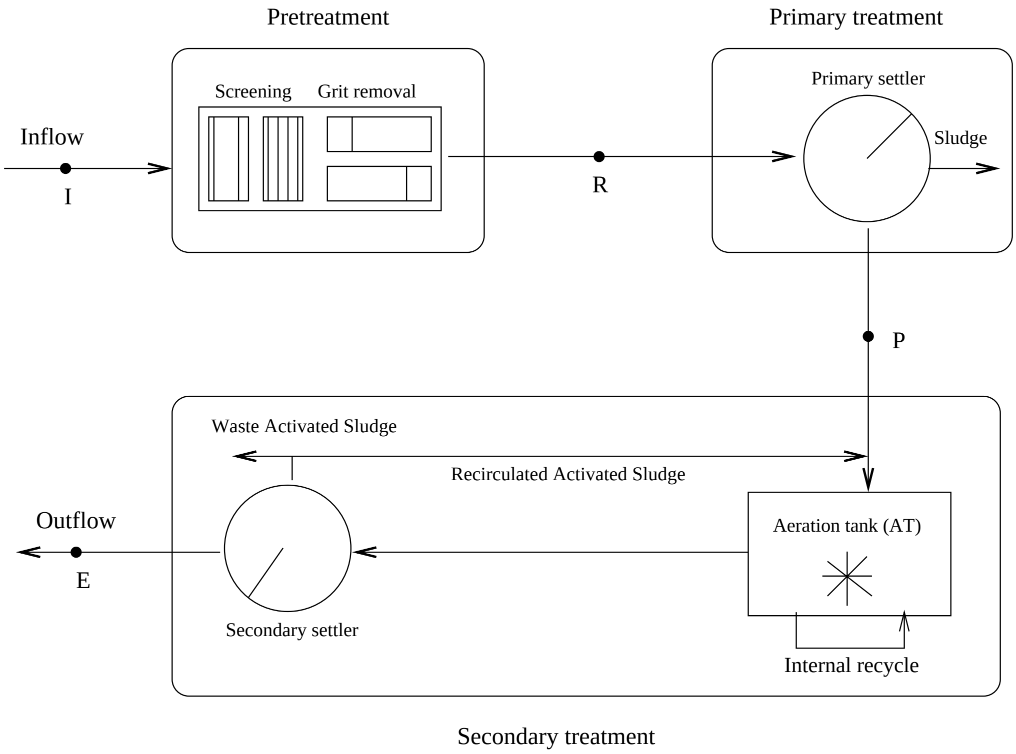

Figure 3.1: Scheme of a waste water treatment plant with activated sludge.

apart from water (out of the system); in this way part of the organic matter associated with these solids is also removed.

- Secondary treatment: the wastewater flows into an aerated tank where *biological treatment* takes place; the micro-organisms of the activated sludge (see §3.2.3 and §3.2.4 for explanations on activated sludge) come into contact with the organic matter dissolved in the wastewater and use it to grow. The organic matter is degraded and transformed in biomass; new activated sludge forms and it is poured into a secondary settler (clarifier), where suspended biomass sediments in form of sludge and is separated from the water, which now is clean and can be released to the external environment.

- Sludge cycle: part of the concentrated sludge (RAS, Recycled Activated Sludge) settled in the clarifier is recycled back to the aeration tank to preserve an optimum concentration of micro-organisms there, while the rest of the sludge

(WAS, Waste Activated Sludge) exits the system. The WAS, after thickening, stabilization and dehydration to reduce volume, can sometime be used as compost.

With activated-sludge treatment plants it is normally possible to meet the general requirements regarding the final quality the water should have. In many cases, however, WWTP standard management wastes resources and energy, and is not cost effective in reaching acceptable effluent[4] quality levels (Wen and Vassiliadis 1998). Moreover, in case of exceptional events (e.g., a storm, a heat wave or problems of biological origin such as bulking and foaming), it can be very difficult to maintain the quality of the effluent water. Specially in case of biological problems, a considerable time will be necessary to return to have a balanced microbiological population.

That is why the application of AI techniques and advanced water treatment technology coupled with on-line quality control techniques in wastewater management is needed. The integration of different AI models could provide an innovative way to:

- improve the efficiency of the treatment process;
- eliminate chemical pollutants;
- reduce energy and chemicals consumption;
- reduce the formation of disinfection by-products.

Local legislation usually regulates all kind of dumping (including the quality of the effluent from wastewater treatment facilities) and defines the maximum allowed concentrations of every single component released from a WWTP into the receiving water body. Regarding this point, the introduction of improved AI decision-support techniques can help to manage complex problems, such as stabilizing the

[4] With the term *effluent*, without additional specification, we refer to the effluent exiting the WWTP. This is different from *primary effluent*, which is the effluent exiting the primary treatment.

composition of the effluent, preventing microbiological pollution, minimizing general environmental-impacts from wastewater treatment outflows, and evaluating the safety of water re-use.

3.2.2 Wastewater characterization

In this section we start to present basic considerations for the management of a WWTP. Essentially, it is necessary to know the characteristics of the raw wastewater and the normal variations associated with it. Then it is necessary to define the acceptable characteristics of the end product and their admissible variations (Benjes 1980).

Water is said to be contaminated when its characteristics deviate from allowable values. Raw wastewater is always contaminated. The contamination can be produced by toxic substances or by non biodegradable materials, but also by natural substances, which, if poured in large quantities, cannot be *metabolized* by the water body, usually a river.

Not all wastewaters have the same composition and the technology for their treatment is different in each case. Wastewater can be characterized in accordance with its origin: (1) domestic or municipal, (2) industrial, (3) agricultural, (4) related to mine drainage, and (5) related to livestock production operations. A specially important feature is the presence of pathogenic organisms, which can prejudice a possible alternative reuse of treated water, such as irrigation.

Raw-wastewater characteristics. A possible classification of the physical, chemical and biological descriptors[5] of wastewater is presented in Table §3.1 (U.S. EPA 1977).

Domestic wastewaters have a rather constant composition, with substantially two constituents: human metabolic waste and discarded material. While the first

[5]See § 5.3.2 on page 114.

Table 3.1: Classification of the physical, chemical and biological descriptors of wastewater.

Physical	**Chemical**	**Biological**
Suspended solids	*Organic*	Protists
Temperature	Proteins	Viruses
Color	Carbohydrates	Vegetables
Odor	Fats, oils	Animals
	Surface-active agents	Pathogens
	Phenols	
	Pesticides	
	Restaurant grease	
	Inorganic	
	pH	
	Chlorides	
	Alkalinity	
	Nitrogen	
	Phosphorus	
	Sulfur	
	Oxygen	

component is almost changeless in nature (as dependent on human metabolism), the second one depends on many parameters, such as standard of living, local habits and country.

Industrial and *mine-drainage* wastewaters are very variable, and have to be treated in a special way and, possibly, at the source. They are waters which should not be mixed with domestic waters, at least not before eliminating their contaminants.

Livestock-production and *agricultural* sectors are often the source of strong contamination due to important amounts of organic matter, purine, nitrates and pesticides remains.

Therefore, when wastewater enters a WWTP, it always contains a complex array of waste materials. This waste is typically categorized into descriptors, which sometimes reflect the specific waste element and some other times reflect the effect of the waste category. The general characteristics and defining descriptors of these categories are summarized below.

Total solids can be distinguished in suspended, colloidal and dissolved, and contain organic and inorganic portions. The size of the solids that are present in wastewater influences the sedimentation, adsorption, diffusion, mass transfer and biochemical reactions.

The *temperature* of wastewater depends on the typology of dumping and on the permanence time in the sewers. Except for summer months, it is higher than environment temperature, due to the presence of warm water dumping from kitchens and bathrooms. The importance of wastewater temperature is bound to the biological activity of purification in treatment plants. At more than 40°C nitrification halts and temperatures higher than 50°C block aerobic digestion. Temperatures lower than 15°C inhibit some anaerobic process, while at 5°C the nitrifying autotrophic flora stops its activity and at 2°C also the heterotrophic flora become ineffective.

Wastewater *color* is strictly correlated to its age, its septic conditions and to the presence of industrial dumping.

The *odor* is associated to putrescence and decomposition degree of organic matter, and to the presence of particular industrial wastewater.

Organic matter is, in general, easily biodegraded. To evaluate its content, the *biochemical oxygen demand* (BOD)[6] and the *chemical oxygen demand* (COD) are determined.

The majority of toxic effects on WWTP-micro-organisms' growth are attributable to *inorganic matter*, such as *heavy metals*, and to its interaction with other wastewater materials.

The *nitrogen* found in wastewater is of five prevalent kinds: organic nitrogen

[6] The BOD represents the amount of oxygen needed by bacteria to degrade the organic matter and it is function of the biodegradable organic matter.

(in vegetal and animal proteins), ammoniacal nitrogen, nitrites, nitrates and elemental gaseous nitrogen. Ammoniacal nitrogen is produced during the decomposition/hydrolysis of organic nitrogen and can come from the bacterial reduction of nitrites or directly from industrial dumping.

The main kinds of *phosphorus* existing in wastewater are: salts of orthophosphoric acid, poly-phosphates and organic phosphorus. In urban wastewater, in general, all kinds of phosphorus are present, while, after a biological treatment, there are generally only ortho-phosphates.

Sulfur is present in the form of sulfates or sulfides. Sulfates can be reduced to sulfides by sulfate-reducer bacteria in anaerobic conditions. Sulfites constitute a culture medium for several species of aerobic bacteria able to create sulfuric acid, which can cause corrosion problems.

Chlorides have metabolic human origin (as they are contained in urine in an amount equal to 1%) or are due to industrial-water contribution.

Some *heavy metals* in wastewater are necessary in minimum amounts as micro-elements for WWTP micro-organisms and for aquatic life, but they are poisonous in high concentrations.

A basic knowledge about the most common natural organisms that can be found in wastewater is also necessary to control the treatment process. Some of these organisms are essential for certain pollution-removal treatments, such as activated sludge. The majority of pathogenic organisms are part of human intestinal bacterial flora and they cannot survive for a long time in wastewater. In general, most of the organisms of human origin are banal saprophytic bacteria, that is organic-matter demolishers; they are not pathogenic and can enter biological processes without any problem (Damiani 1991).

Variations in wastewater characteristics. Wastewater-treatment facilities are not designed for average wastewater characteristics. Only the most unsophisticated design would provide treatment capacity for the average flow and the associated average characteristics. Treatment facilities have to be designed to accept and treat the peak wastewater flows and associated characteristics. Therefore, it is important to identify the variations in wastewater characteristics and to analyze these variations. Changes in wastewater descriptors during the day and during the year are taken into account. The analysis of these changes permits the estimation of other important descriptors in activated-sludge processes by use of kinetic models. For example, the data of the BOD to aeration ratio allow the assessment of peak oxygen-uptake rates in activated-sludge. The plant process design is a function of peak loading conditions, not average conditions (Benjes 1980). Management applications have to be able to achieve acceptable effluent quality during peak loading conditions, too; therefore we have to be aware of the variations in wastewater characteristics, and design the system taking into account the variation in loading and above all the maximum loading.

Effluent requirements. Effluent requirements are a legally binding commitment. The requirements are usually stated in terms of BOD and *total suspended solids* (TSS) (monthly average, 7-day average, 24-h composite, and grab sample). Some of the effluent criteria permit variations from an average performance value, but the grab-sample values are never to be exceeded. The monthly average value for BOD and suspended solids represents a flow-weighted numerical average (Benjes 1980).

3.2.3 Aerobic biological waste-treatment

Aerobic biological waste treatment can work by suspended growth (activated sludge) or attached growth (trickling filter). In both cases it follows basic concepts. Very fundamentally, the process converts raw waste organic-matter to bacterial cells, which are subsequently separated from the liquid stream. This requires a medium for bacterial growth and oxygen for organic conversion to cells. We focus on WWTPs which have an aerobic biological reactor (the unit that contains the activated sludge). The choice of aerobic biological processes is appropriate for the treatment of urban wastewaters, due to their origin and composition. This choice is also valid for some industrial wastewaters, whose treatment however requires a special preliminary study.

3.2.4 Suspended-growth systems

The activated-sludge consists in a mixed microbial culture which grows aerobically on the (organic and inorganic) components of wastewater, producing new micro-organisms which are separated, in the settlers, from treated water. This ability to settle and separate from treated water is due to the basic characteristic of *formation of flocs*. All micro-organisms show common nutritional demands: they need a carbon source from which to obtain energy to grow and reproduce; besides, they need nitrogen, phosphorus, potassium, oxygen and water; and finally they can show specific demands for different chemical elements and organic compounds (Ca, Mg, Fe, S, vitamins and hormones). The final products of micro-organism activity, besides new biomass, are: catabolism by-products, water and carbon dioxide.

To control the process of the activated-sludge unit, knowledge about the biological and chemical characteristics of the water and about the micro-organisms of the sludge is needed. It is also important to have some knowledge about urban washing and rain water, because they contribute a lot of pollutants, and high concentrations of toxic polluting-substances in the WWTP are able to prejudice the efficiency of the treatment and the equilibrium of the sludge environment.

All biological processes in the activated-sludge reactor are characterized by the nature of the substrates, by the kind of micro-organisms and by the final products that form. In the biological treatment of a WWTP, the demolition of the organic matter (substrate) contained in the wastewater by the micro-organisms takes place

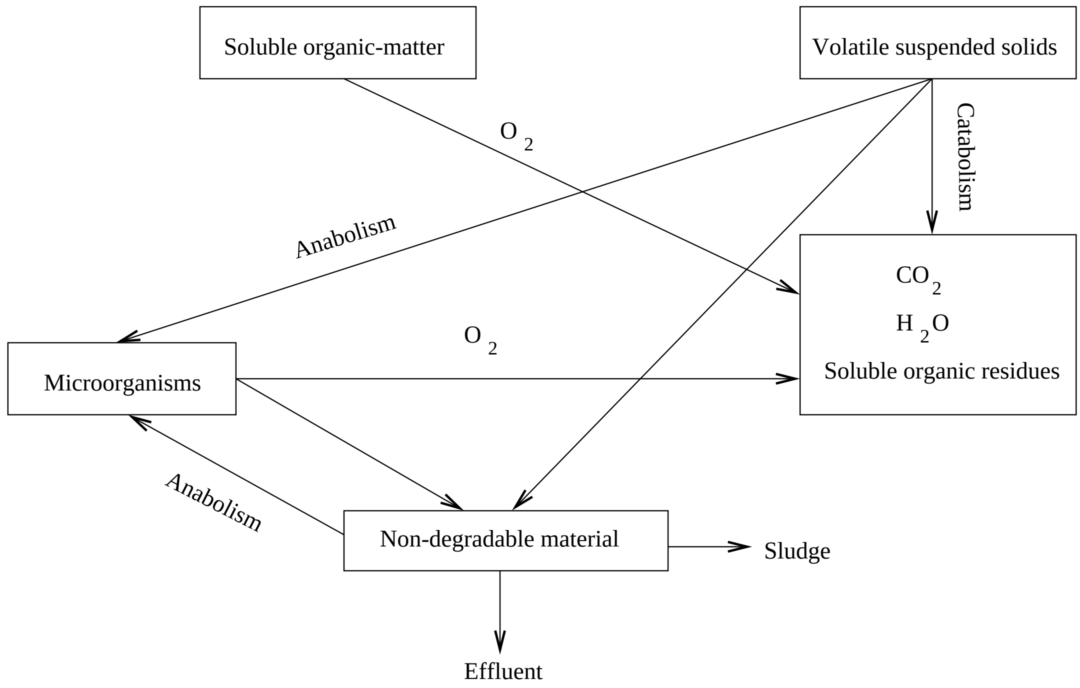

Figure 3.2: Aeration-tank processes.

according to processes analogous to the ones of water-stream auto-purification, but in such a way as to make the transformations occur with much higher rate and performance. The cellular metabolism of WWTP-micro-organisms is made up of two phases (see Figure § 3.2, modified from (Jørgensen and Johnsen 1989)):

- anabolism: constructive phase of the metabolism in which micro-organisms transform the substrate into the material that constitutes the protoplasm and the reserve substances;

- catabolism: transformation, generally destructive, of the substrate, with release of the necessary energy for the synthesis and cellular preservation processes.

The biochemical activity exerted by microbial cells on the substrate takes place through enzymes produced by the very cells. These enzymes are in part discharged into the extra-cellular environment and in part held in the cells or on the cells.

The enzymes take part in the reactions which govern energy freeing and protoplasmic synthesis; the transformation of the organic matter constituting the substrate occurs through the formation and subsequent decomposition of enzyme-substrate complexes.

3.2.5 Micro-organisms in WWTPs

The metabolic activity of WWTP-micro-organisms is very high, but they are also very sensitive to the parameters that affect cell reactivity and kinetics (temperature, radiation, pressure, pH and substrate concentration). In the thesis, we will not go into the detail of the kinetics of the biological reactions, firstly modelled by Monod (1942), but we will deal with the relation among the presence of certain micro-organisms, the running of the biological purification process and the state of the WWTP.

To evaluate the presence of micro-organisms, a periodic microscopic observation of the activated sludge is essential. Some microbial populations, such as filamentous bacteria (e.g. *Actinomycetes*) that produce biological foams or activated-sludge bulking, are problematic and have a negative influence on wastewater treatment if they are in excess. On the other hand their presence in equilibrium with the floc-forming bacteria is necessary for the optimum floc formation. Other microbial populations, such as bio-P bacteria (or phosphorus-accumulating bacteria), are important for their positive role. Now, even if various kinds of micro-organisms can be found in a WWTP, the biodegradation activity is mostly carried out by bacteria (although protozoa and small metazoans[7] can contribute to effluent pureness), to which the prokaryotic nature gives extreme versatility in the use of a wide range of substrates and very reduced duplication times. In the aerobic processes, aerobic and facultative-anaerobic bacteria are active, while, in strictly anaerobic processes,

[7]Animal with specialized cells.

facultative and obligatory anaerobic bacteria (like the ones involved in methane formation) are active. Moreover, in plants with special nutrient removal units, there are also aerobic bacteria with the possibility of anaerobic respiration (denitrifying bacteria, which carry out, in the anoxic reactors, the nitrate reduction).

The microbial composition of the activated-sludge (a micro-community in the form of biological flocks) and its activity depend on many factors: kinetic constants of the different species, availability and nature of the carbon substrate and physical conditions (pH, temperature, dissolved oxygen, mixing system, plant configuration). Besides, it is necessary to consider also the interactions among different components of the micro-community, such as predation, commensalism and food competition.

While simple visual microscopic observations give sufficient information on activated-sludge microbial populations, in anaerobic biological treatment the main tool of characterization of anaerobic micro-organisms are estimations by means of microbiological analyses (which are more difficult and expensive) of the most important microbial populations.

Considering the variability of several parameters which are involved and the complexity of the activated-sludge system, it is a very arduous task to reach an equilibrium and to keep steady the composition of biological aggregates while guaranteeing specific results in terms of purification. Indeed, despite the progress in the study of the microbial ecology of activated sludge we still do not have a complete knowledge of the interventions to be adopted when problems like bulking[8] or foam abnormal-production (foaming) arise.

With the aid of suitable AI systems, including an ontology, the automatic association of the presence and evolution of micro-organism communities with the control and management of WWTPs seems to be feasible. For example, it is possible to identify, on a morphological basis, numerous filamentous-bacteria which cause

[8]Situation of poorly settling sludge.

bulking or foaming problems and with such information, even having a limited case history, to outline an operational modification of the plant in order to restrain the growth of the responsible microbial species and then to recover from the malfunction possibly generated (Tandoi 1991).

3.2.6 Process performance

The characteristic capability and reliability of various processes are important considerations in meeting effluent criteria. Not only is the average effluent quality important, but the extremes have to be considered to assure meeting the criteria imposed on most plants (Benjes 1980).

The *activated-sludge* process has the capability of converting essentially all inflow soluble organic matter to solids. It is then necessary to remove the solids efficiently in order to attain high-quality effluents in terms of organic matter. When dealing with large input quantities of solids and problems of sedimentation or thickening, careful operational consideration of solids balances is necessary to attain consistently good effluent quality.

Several plants (not in Catalonia) are required to produce an effluent having limitations on ammonia nitrogen, because a high level of ammonia is one of the main problems in many rivers. Biological nitrification is one of the least expensive methods to achieve low concentrations of ammonia nitrogen. The introduction of nitrification also affects the whole process performance.

3.2.7 Sampling, instrumentation and control

Sampling points are in general provided before and after each unit process. The frequency of sampling is determined by the operating agency. Good points of sampling are critical to assure that representative samples can be obtained. In many plants, continuous sampling is practiced by installing small pumps to deliver continuous samples to the laboratory. In this case, long times in transit may affect the quality of the sample (Benjes 1980).

A variety of automatic sampling units are available, too. Each unit obtains a representative sample of the flow during the period in which the sample is taken. Generally, a stream is continuously pumped from the sample point past the sampler. Suitable types of samplers permit taking samples at intervals of a constant amount of flow. In general, automatic samplers which require dipping through the surface of the wastewater are avoided. Floatable materials tend to accumulate at the surface in greater concentration than in the total body of the fluid and thus dipping may give an unrepresentative sample. Most of the automatic samplers can be equipped with refrigerated cabinets designed to maintain the sample at the suitable temperature range to preserve it in an approximately unaltered condition.

Properly designed and adequately maintained automatic sampling equipment provides more precise information of plant performance than manual sampling, because samples are obtained more frequently and more constantly. In any case, automatic samplers have to be checked routinely by hand sampling.

Instrumentation and control equipment adapts to the needs of a particular plant. Which data generated are for long-term planning, which data are of value to the operator and which ones are to be used in automatic control have to be determined. The plant's operation is based on the flow, oxygen demand and effluent quality that is occurring at any given time, and the experienced response from preexisting situations (Benjes 1980).

The control equipment, including decision-support systems, is primarily used by the operator. In general, non frequently used controls are distributed locally, whereas continuously used controls are located in the control room or performed automatically. For instance, inflow pumping operations can be automatically controlled based on the flow rate.

3.3 Classic control of the activated-sludge process

In this section and in chapter 4 we describe various techniques used for the control, supervision and optimization of WWTPs.

The first control approaches are in general simple and based on the hypothesis of a steady-state system. This is due to the fact that, to progress in the understanding of wastewater treatment processes, it was thought to be practical to begin with simplified models and not to take into account that the real world is more complex. The real world is always more complex (Ball 1984), it was thought.

In the classic control of activated-sludge systems, the basic descriptors on which it is possible to operate are just three:

- air or oxygen flow;
- sludge return-flow;
- waste-sludge purge-flow.

The basic control strategies used in the classic approach are:

1. numerical modelling for control (e.g., Environmental Tracing Systems Ltd., http://www.environmentaltracing.com/wastewater.htm);
2. establishment of the relationships among different state descriptors (e.g., different kinds of substrates and micro-organisms);
3. use of adaptive algorithms for maintaining a desired DO level in the bioreactor;
4. simulation of the evolution of the state of the treatment plant;
5. use of activated-sludge models to estimate the value of the state descriptors that characterize some process, but cannot be measured on line (e.g., substrate and biomass concentration);

6. descriptors estimation for fault detection;

7. interactive methodology with the plant's manager proposing actions and the system advising about the likely answer of the plant processes according to a previously calibrated model;

8. classification of sensor data through fuzzy clustering procedures (Marsili-Libelli 1998).

The application of these simple control strategies and simulation enables the achievement of the following specific and general benefits:

- evaluation of WWTP's behavior in response to certain scenarios (operational conditions and inflow water's composition) and prediction, in the medium and long term, of likely consequences of alternative actions taken over the process;

- contribution to the study of alternatives for upgrading or retrofitting of existing WWTPs;

- increase of the average efficiency in pollutant removal;

- more constant characteristics in time of the output effluent;

- reduction of energy, chemical reagents and staff costs.

Moreover there is an aspect which helps WWTP control and it is related to the different order of magnitude in the response time of the parameters used (see Table § 3.2 on the next page). This fact avoids interference and can help the integration of various simple control-actions (Olsson and Newell 1999). In all cases, to carry out control actions effectively, it is very important that all necessary calculations can be executed in real time and that the input feed of short response-time parameters is received continuously (Tomei and Di Pinto 1991).

Table 3.2: Orders of magnitude of response time for some fundamental parameters.

Parameter	**Response-time**
Air flow	Minutes
Sludge return-flow	Hours
Waste-sludge purge-flow	Days

3.3.1 Classic control's problems

In spite of all the advantages noted above, the scientific community has realized that the use of numerical models, on which the classic approach for control is based, presents many limitations. Since the process of wastewater treatment is very complex, to develop a reliable supervisory technology based only on chemical-engineering control can be considered a good try, but it did not solve all the problems. Indeed, the application of classic, automatic process-control to wastewater treatment systems has shown several difficulties, related to the following issues:

- Complex cybernetics of the system (Wiener 1961): as in every environmental system, there are many external factors and a lot of internal feedback-signals influencing the wastewater treatment process, such as the interactions among the many micro-organisms coexisting in the biological reactor or between these micro-organisms and the substrate. This complicated cybernetics makes difficult the establishment of valid models for the description of the behavior of the system. The results of simulation by numerical models are only valid when applied appropriately, i.e. when they are applied to a plant behaving accordingly to situations that have been considered during the calibration phase. Simulation cannot deal with unknown or non-modelled situations. Models are often not easy to be developed, and they are frequently inaccurate and excessively simplistic representation of reality.

- Qualitative information: a lot of data describing the processes (such as water color and smell, predominant protozoan, presence of bubbles in the V30 settling test or the flocculation state) and the problems (e.g., filamentous bacteria proliferation, bulking, rising) cannot be numerically quantified and therefore cannot be used in the context of a conventional numerical-control model. It is indeed considered unsatisfactory the modelling of such problems without taking into account qualitative information.

- Uncertainty: the knowledge about many descriptors of the process, such as BOD and volatile solids, is approximate and difficult or impossible to obtain on-line. Even when data are obtained on-line, *a sensor can fail*, and the supervisory system receives noisy data or no data. Moreover, input information is affected by missing data, because not all descriptors are analyzed every day. Therefore, some subjective information, based on experience, has to be introduced by human experts and to be taken into account to identify with precision the state of the plant.

- System dynamics: the system is under continuous change (several parameters of the system, such as the wastewater-flow composition and magnitude, are very variable in time) and this circumstance modifies the performance of the process over time. In contrast with most industrial processes, where type and amount of raw input material are under control, WWTPs receive a wastewater flow whose magnitude and pollutants concentration are definitely variable and uncontrollable. Therefore, a real-time control loop is needed to supervise the process.

- Delays in data capture: information arrives at the control system with different delays with respect to the sampling time. On the one hand, a few descriptors are measured on-line; on the other hand, some of them can be subjected to important analytic delays (e.g., hours up to days in the case of BOD).

- Reliability: important maintenance efforts are necessary for the reliable functioning of on-line analyzers (for ammonia, nitrite, nitrate and dissolved oxygen), on whose data models are based.

It has been seen that classic-control methods work well when the plant is in a *normal* state, but they do not if the state of the WWTP is *abnormal* [9]. It is clear that, if the skills of an expert engineer or technician need to be captured, far more than building bigger simulators or equation solvers has to be done; the computer system has to somehow embody the common sense of these experts. This common sense includes:

- detecting completely new situations (e.g., a toxic-substance shock);
- using the subjective information accumulated through years of experience;
- using incomplete information and the objective information provided by years of continuous WWTP operation.

In next chapter we will see how AI control-systems have shown to be able (1) to cope with some of the difficulties of conventional process-control, and (2) to improve theoretically and practically the management of several real-world problems.

[9] An *abnormal* situation refers to the incorrect operation of a WWTP. This means that some of the pollutants' concentrations at the effluent do not fulfill the legal constraints (and therefore that the treatment goals are not being achieved), or that some of the main operational descriptors or inflow characteristics are not within normal ranges.

Chapter 4

Environmental Decision Support Systems

Think hunger is no longer a threat?

Dream on.

When dealing with complex environmental problems, with managers who may not have sufficient knowledge of environmental issues, or with environmental processes which are not easily modelled because our knowledge is still incomplete and uncertain (Cortés *et al.* 2001), EDSSs can be useful.

EDSSs are instances of *decision support systems* (DSSs). An EDSS is the integration of KBSs, applied to an environmental issue, that potentially reduces the time in which decisions are made and improves the consistency and quality of those decisions (Guariso and Werthner 1989). Advancements in DSSs research have the potential of benefiting many environmental fields (thus the creation of the EDSS research category), as well as very different disciplines. These advancements are in general results in applied optimization, process-control improvements and environmental decision-making.

From the user point of view, an EDSS facilitates an iterative decision-making process, in which the decision-maker/analyst incrementally learns more about a problem during the supported management process. As new bits of knowledge are gained, previous assumptions may be challenged. Through a number of iterations, one or more solutions to the problem can be incrementally developed and refined. In addition, a DSS simplifies the supervisory process by insulating the user from tedious tasks, such as accessing data and setting up modelling studies. This allows the user to concentrate on developing, testing and comparing alternative control strategies.

A potentially important component of such systems is continuous improvement, of which optimization is an instance. Continuous improvement allows the DSS to take a more proactive role in the supervisory process by generating low cost alternatives that meet user-defined goals and objectives. These alternatives may be good starting places for analysis and may provide the user with valuable insights about how to efficiently solve a problem.

While this vision of an environmental management DSS can be readily articulated, implementations that fit this description are rare. One reason is the complexity of many problems faced in environmental management. For example, developing water-quality management strategies for wastewater treatment includes the following difficulties: large number of potential control options, but no control on wastewater sources; uncertainties in water composition, process control and meteorological data; complex pollutant behavior; time-consuming process simulation models.

DSS features can be designed to facilitate consideration of these issues; however, the long runtime of simulation models makes identifying feasible control strategies difficult. This may make the ultimate goal of generating and comparing management alternatives impractical. Furthermore, the complex behavior of many environmental problems cannot be easily incorporated into traditional control approaches. Non traditional approaches, such as rule-based reasoning, case-based reasoning, ontologies, *genetic algorithms* (GAs) (e.g., Loughlin (1998)) and simulated annealing (SA)[1] can be used instead, but the computational intensity of some of these approaches may

[1] *Simulated annealing* is a very general optimization method which stochastically simulates the slow cooling of a physical system.

limit their applicability.

An advantage of GAs and SA is their ability to accommodate complex environmental-process models into an optimization search process. While this is an important characteristic of GA and SA search, the process model may be executed up to 20,000 times. If the process model requires only about a second to run, the optimization run would require several hours, that is acceptable. However, if each run of the process model requires 1 minute, the optimization runtime would be up to 10 days. This duration is not practical in most realistic decision-making scenarios, but it can be made practical using parallel versions of GAs and SA. In this thesis, we do not explore how distributed high-performance computing can improve optimization of this type, considering the use of parallel computers unrealistic within most environment-related processes, such as wastewater treatment.

In chapter §2 we analyzed the role of ontologies in knowledge reuse and *engineer to engineer* (E2E) Web-based collaboration. In this chapter, we start exploring two other kinds of approaches: rule-based expert systems and case-based reasoning systems. These are among the most used components in EDSSs. Then we discuss which features an EDSS should include, and which kind of knowledge and problems an EDSS should be able to deal with.

4.1 Rule-based expert systems

Rule-Based Expert Systems (RBESs) are advanced computer programs which emulate, or try to, the human reasoning and problem-solving capabilities, using the same knowledge sources, within a particular discipline (González and Dankel 1993; Jackson 1990; Buchanan and Smith 1988; Clancey 1985; Hayes-Roth 1984; Stefik *et al.* 1982). RBESs always possess certain heuristics that form the static knowledge-base, and some inference and search processes. The problems addressed with RBESs are very complex and related to specific domains, and they would usually need a very expert human (i.e., a large amount of knowledge) to be solved[2]. A few examples of real-world applications of RBESs to environmental issues are the following ones:

1. decision support for natural resources management (Fedra 1995);

2. data management in forestry (Matwin *et al.* 1995);

3. petrochemical-plant control (Alamán *et al.* 1992);

4. dynamic-process monitoring and diagnosis (Finch *et al.* 1990);

5. WWTP time-series analysis (Novotny *et al.* 1990);

6. control of sun-powered systems (Sanz *et al.* 1988).

The main components of RBESs are: static knowledge-base (or long-term memory), data base (or working memory or short-term memory), inference engine, user interface, auto-explanation module, strategy module, knowledge-engineer interface and on-line sensor/effectors interface (see Figure §4.1).

Typically, the knowledge contained in the historical data is encoded in the static knowledge-base in the form of rules or axioms, by way of a knowledge-acquisition process such as the one described in Figure §4.2. The rules allow the system to deduce new results from an initial set of data (premises). A rule is basically constituted by the following structure:

IF *conditions* THEN *actions*

The reasoning method (inference engine) may use forward chaining, backward chaining or a combination of both of them. Forward-chaining reasoning starts from the input data towards the final conclusions, deducing new facts from previous ones. Backward-chaining reasoning is guided by the conclusions towards the input data (commonly provided by the user).

[2]It may even happen that the RBES algorithmic-power could do some special tasks that the human one (the mind) cannot do in the great majority of the cases.

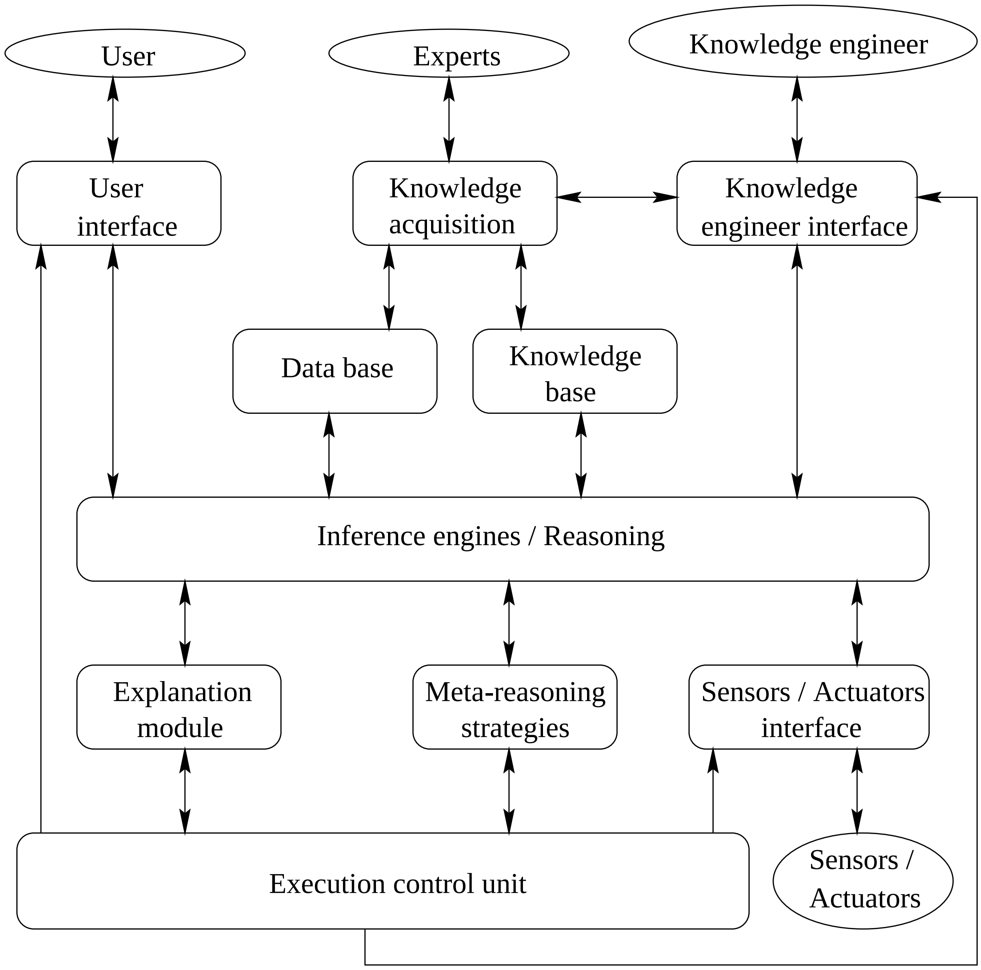

Figure 4.1: A scheme of an expert system.

Important issues, which contribute to making RBESs very useful, are the following ones:

- When applied to a specific domain in which experts can provide experience and knowledge, RBESs are very effective in assimilating this expertise.
- The separation of the static knowledge-base from the control elements provides an easier way of building and updating RBESs.
- RBESs are highly interactive.
- RBESs support numerical and symbolic information.
- RBESs can manage ill-structured domains and approximate reasoning.

Thanks to their characteristics, RBESs have been widely and successfully applied to environment management, supervision and control (Cortés and Sànchez-Marrè 1998; Sànchez-Marrè 1995; Mason 1995; Dym and Levitt 1991; Stephanopoulos 1990; Guariso and Werthner 1989; Hushon 1987; Sriram and Adey 1986; Efstathiou and Mamdani 1985).

Examples of specific applications of RBESs to the management of environmental problems are the following ones:

- an expert system for water supply operation (Shepherd and Ortolano 1996);
- an expert system for water treatment plant operation (Zhu and Simpson 1996);
- an expert system for advising emergency teams about how to deal with industrial accidents (Avouris 1995);
- expert systems for water resource simulation and optimization (Fedra 1993);
- an expert system used to help in environmental planning (Wright *et al.* 1993).

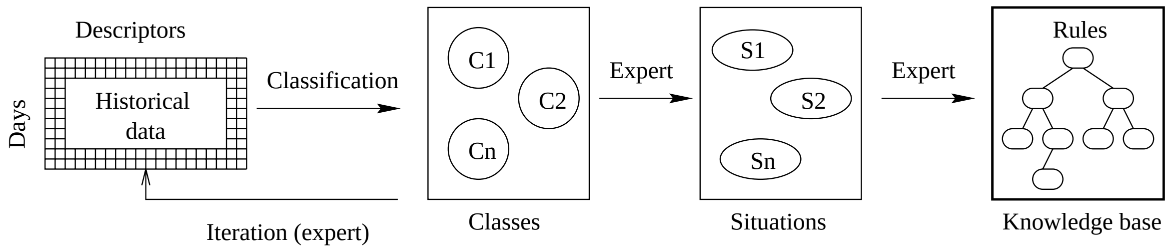

Figure 4.2: Knowledge acquisition in RBESs.

4.1.1 RBES and wastewater environment

Focusing finally on the supervision and control of WWTPs, examples of RBESs application to WWTP are:

- a RBES to supervise a WWTP via a local net or the Internet (Baeza *et al.* 2000);
- a system for automatic construction of rules used to identify WWTP states (Riaño 1998);
- an expert system for selecting and sequencing wastewater treatment processes (Yang and Kao 1996);
- a RBES for control and supervision of urban WWTPs (Serra 1993);
- a RBES for the general operation of wastewater treatment processes (Barnett and Andrews 1990);
- an expert system for sewer network maintenance (Ortolano *et al.* 1990);
- an expert system for dynamic modelling (Patry and Chapman 1989).

4.1.2 RBESs' problems

Despite all the existing applications, rule-based expert systems are not completely satisfactory because they do not incorporate some desired features of human intelligence and could present technical difficulties in being developed. The main problematic issues are:

- Most expert systems *do not learn from their experience*, while this kind of learning would be a valuable feature in many systems (Aamodt 1989).

- There are some difficulties in the process of *extracting knowledge and experience* (see Figure §4.2) from the sources (Becker 1987).

- Most RBESs are *brittle*, their scope being limited to past and forecasted domain-situations. If RBESs are rigidly implemented, i.e. if *there is no user friendly interface* to change the rules, they are not reliable when applied to unexpected circumstances (Steels 1990). Moreover they need a specialized programmer to make changes to the KB. Knowledge and experience constantly change and RBESs need a dynamic behavior. This behavior can be achieved with *shells*, which help in the construction, validation and execution of RBESs. Shells include one or more inference engine, but no KB. Working with shells, knowledge engineers, during the construction of a system, can freely define the KB. They can also easily correct errors in the KB or have more than one KB working with the same inference engine. Shells are then a good means to develop expert systems which can be adapted to new situations.

- The complexity of RBESs increases as the systems themselves grow if they are 'monolithic' architectures, and to manage the information and knowledge contained in them becomes more and more difficult.

- Building new RBESs today usually entails constructing new static knowledge-bases from scratch. Assembling reusable components could be an alternative. RBES developers would then 'only' need to worry about creating the specialized knowledge and reasoners that are new to the specific task of their systems or domains. The new RBESs could inter-operate with existing systems, using them to perform some of the reasoning. In this way declarative knowledge, problem-solving techniques and reasoning services could all be shared among systems. This approach would facilitate building bigger and better systems cheaply. The infrastructure to support such sharing and reuse would lead to greater diffusion of these expert systems, potentially transforming the knowledge industry (Neches *et al.* 1991). We already dealt with knowledge sharing and reuse in §2.4.9.

In the next section we will see how case-based reasoning can help to solve several of the problems of RBESs presented and to model specific knowledge in the WWTP domain.

4.2 Experiential knowledge and *case-based reasoning* (CBR)

CBR is both a paradigm for computer-based problem solvers and a model of human cognition. The central idea is that the problem solver reuses the solution from some past case to solve a current problem.

CBR as a computer program paradigm. As a paradigm for problem solvers, one of the advantages of CBR systems is that they improve their performance, becoming more efficient, by recalling old solutions given to similar problems and adapting them to fit the new problems. In this way they do not have to solve new problems from scratch. The memorization of past problems / episodes is integrated with the problem-solving process, which thus requires the access to past experience to improve the system's performance. Additionally, case-based reasoners become more competent during their functioning over time, so that they can derive better solutions when faced with equally or less familiar situations because they do not repeat the same mistakes (learning process). The basic steps in CBR are (see Figure

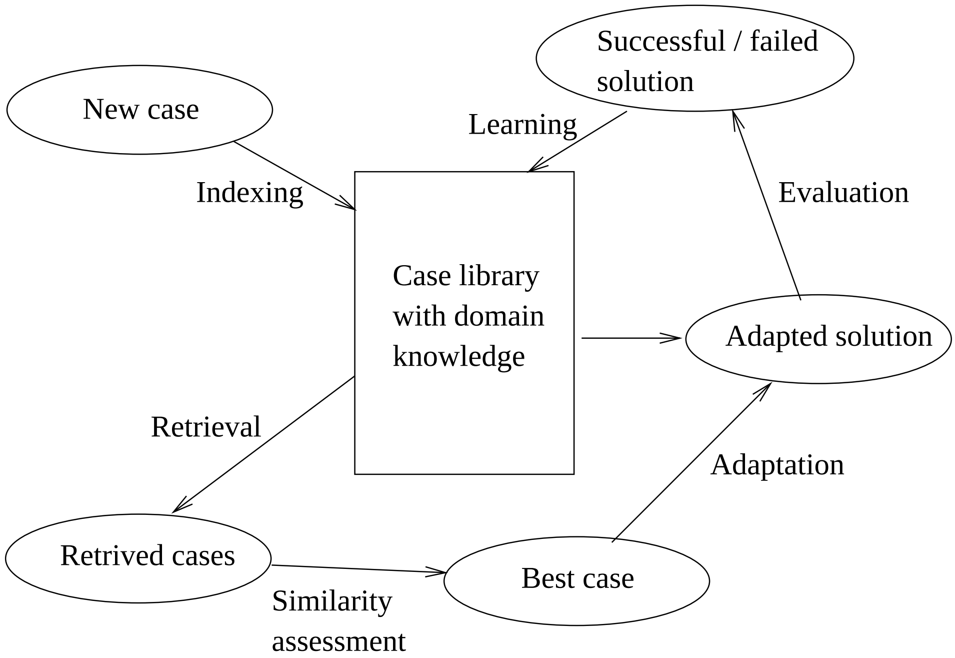

Figure 4.3: The CBR paradigm.

§4.3):

1. Introducing a new problem (or situation) into the system.

2. Retrieving a past case (a problem and solution), whose problem part resembles the current problem. Past cases reside in case memory. The case memory is a library that contains rich descriptions of prior cases stored as units. Retrieving a past case involves determining what features of a problem should be considered when looking for similar cases and how to measure degrees of similarity. These are referred to as the Indexing Problem and the Similarity Assessment Problem.

3. Adapting the past solution to the current situation. Although the past case is similar to the current one, it may not be identical. If not, the past solution may have to be adjusted slightly to account for differences between the two problems. This step is called Case Adaptation.

4. Applying the adapted solution and evaluating the results.

5. Updating the case memory (learning). If the adapted solution works, a new case (composed of the problem just solved and the solution used) can be formed (direct learning). If the solution at first fails, but can be repaired so the failure is avoided, the new case is composed of the problem just solved and the repaired solution. This new case is stored in case memory so that the new solution will be available for retrieval during future problem solving. In this way, the system becomes more competent as it gains experience. Updating case memory can include deleting cases (forgetting), too. This step is also part of the Indexing Problem.

Not all case-based problem solvers use all of the steps. In some, there is no adaptation step; the retrieved solution is already known to be good enough without adaptation. In others, there is no memory update step; the case memory is mature and provides adequate coverage for problems in the domain.

CBR as a model of human cognition. As a model of human cognition, some authors argue that CBR is the basic cognitive process by which people solve problems and get about in the world (Schank 1982). Certainly, we have all had the experience of remembering some past situation when confronted with a problem and of finding that that reminding was helpful in solving the current problem. Whether or not CBR is a universal model of human cognition, there are many situations in which people use CBR to solve problems. For example, designers usually rely on libraries of past designs to solve new design problems. The design process often resembles the 5 steps sketched above.

References to CBR bibliography. To quickly identify relevant references to CBR, a quite complete bibliographic categorization exists until 1994 (Marir and Watson 1994)[3]. Recent research efforts in CBR are then presented periodically

[3] http://online.loyno.edu/cisa494/papers/Marir.html

at the International Joint Conference on AI (IJCAI), the European Conference on AI (ECAI), the American National Conference on AI (AAAI), the International Conference on Case-Based Reasoning (ICCBR) and the European Workshop on Case-Based Reasoning (EWCBR). The proceedings of these conferences are a good picture of the current work in the CBR community.

4.2.1 CBR and wastewater environment

The practical totality of the research about the application of CBR to the WWTP domain is being carried out within the KEML group at UPC (Sànchez-Marrè 1995; Sànchez-Marrè *et al.* 1998) and is focused on plant supervision. Some research about planning and heuristic search have been also accomplished (Krovvidy and Wee 1991), but not recently.

The CBR-related study of the KEML group is part of a more general approach to the modelling of the dynamic learning and adaptation processes needed to accurately supervise and control WWTPs. The specific knowledge supplied by previously solved problems (experiential knowledge) is integrated in the architecture described in Sànchez-Marrè *et al.* (1996), which is the antecedent of the one used in this thesis. Experiential knowledge is modelled by cases, or experiences, that are organized in a case library. Cases stored in the library are real WWTP operating states, which are learned in such a way that it is possible to reemploy them to solve future tasks. A case incorporates the following set of features: an identifier, the situation description, the situation diagnosis, the action plan, the derivation (from where the case has been taken / adapted), the solution result (success / failure), a utility measure, a distance / similarity value. An example of case representation in this domain is (Sànchez-Marrè 1995):

```
(   :identifier Case-18
    :situation-description (        (Q 35.2 m3 d-1)
                                    (I-COD 289 mg L-1)
                                    ...)
    :diagnostics Normal-Situation
    :action-plan (                  (1 Maintain-the-numerical-
                                    -control-algorithm)
                                    (2 Adjust-Dissolved-Oxygen-
                                    -Value)
                                    ...)
    :case-derivation Case-13
    :solution-result Success
    :utility-measure 0.7
    :distance-to-case 0.378)
```

Case retrieval is standard, with the recall from the case library of the cases which are most similar to the current case. A new distance function is defined for this task. Case adaptation follows and related adapted solutions are derived. With time, the performance of the CBR system is improved; the system becomes *more efficient* by recalling old solutions given to similar problems and adapting them to fit a new problem, rather than having to solve the new problem from scratch. Additionally, the system becomes *more competent* in its evolution over time, because it can derive better solutions when faced with poorly experienced situation. This is accomplished avoiding to repeat the same mistakes and can be considered a learning capability.

4.2.2 CBR's problems

In general, case-based reasoning proved to be a good choice for experiential-knowledge (specific-knowledge) management. But CBR has the basic problem that it cannot work alone if there is no available experience, such as in the case of the initial running period of a treatment plant. It has to be combined, for instance, with a rule-based or an ontology-based system (general-knowledge managers) so that it can work as a reasoning component in the overall control and supervision of WWTPs. An integration of different AI methods is needed, that includes the management of qualitative information (e.g. microbiological descriptors, in the case of wastewater treatment), experts' intelligence and experiential knowledge. In next sections, we will see how DSSs feature this kind of integration.

4.3 Features of EDSSs

Every EDSS should include the following features:

User friendliness. All modules of an EDSS (whose general scheme is shown in Figure §4.4) should present the analysis' results in a user-friendly manner. A few rules in this sense are: not to give the user unneeded choices; not to assume the user knows much about the domain; to remind the user the meaning of the choices[4]; progress indicator bars make users happy; to store information when the user provides it (in any way) and not to ask for that information again.

Assistance in problem formulation. The user is assisted in deciding which objectives need to be reached in each particular moment, when and how the different available tools have to be applied, how to control system's resources if they are limited, how to formally state the dynamic decomposition of a problem, how to organize information.

Framework for information capture. A structured framework draws information, in a logical manner, from the user and the environmental system, about

[4] This is what should not happen: ≪Two choices are presented: *AS plant* and *PH-CH plant*. Since nothing is said about whether to choose AS or CH-PH, the user picks AS, for no particular reason. And he chooses *Use Graphical* login (*Graphical* is a word beginners like). And he decides to *Skip ASM1 configuration* since he does not even know what it is and it sounds very advanced and scary.≫

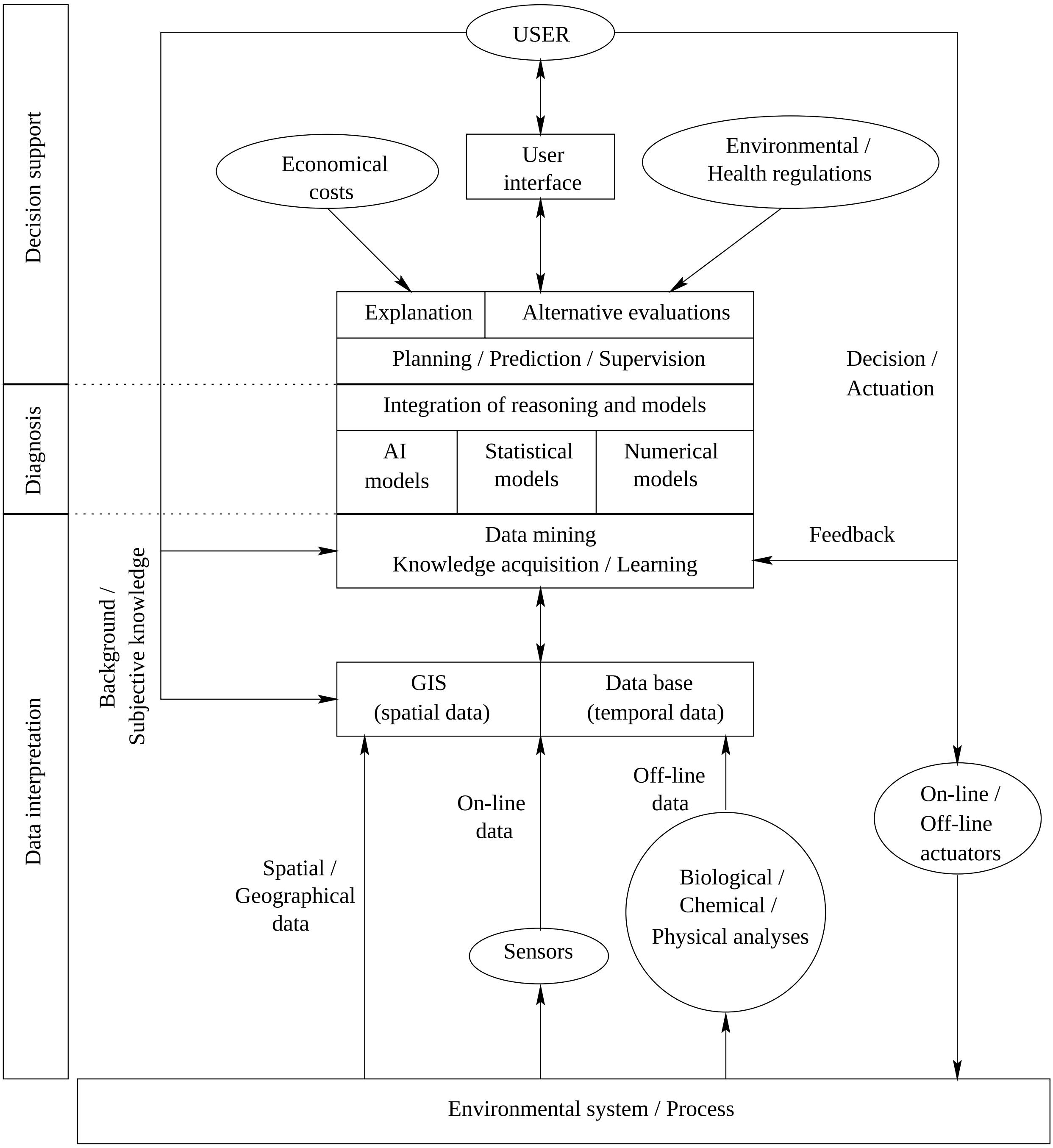

Figure 4.4: General scheme of an environmental decision-support system.

domain-characteristics and processes. This framework, besides acquiring the domain knowledge, has to be able to organize and represent it.

Specific knowledge-bases. These knowledge-bases are associated to the type of domain being considered or to the process being carried out at the site. They contain data on environmental parameters and processes that are relevant to the domain, for instance: what processes are required to manufacture a particular product; what toxic materials are used in the processes; which kinds of physical, chemical and biological samples need to be collected; which is the relative importance of the features in play; which are the requirements of the local legislation.

Integration of different AI methods. It can be considered evident that for the solution of complex tasks it is beneficial to make use of combined methods and procedures. Different applicable methods can have distinct time-limitations, and can lead to solutions with diverse quality and computation-resources needs, allowing a richer cost-benefit analysis and a more reliable management.

Generation of different alternative strategies. The EDSS generates different alternative solutions (according to different environmental definitions/visions) when a problem is met. It is important for alternatives to be very distinct from each others and not variations of a single action schema.

Evaluation of alternatives. General environmental knowledge is used to deduce/assess the relative significance of environmental impacts of different alternatives, which are tested to assist decision-makers in interpreting the results and eventually choosing the most appropriate solution. The evaluation should be on a quantified basis. There are a number of techniques available for this purpose: *check list of* *criteria*, which involves qualitative statements about the extent to which broad objectives are met; *goals achievement matrix* (GAM), which involves quantified statements about achievement of specific objectives using measurable criteria; *social cost benefit analysis* (SCBA), which involves full cost/benefit analysis for all aspects of a plan; *planning balance sheet analysis* (PBSA) which involves cost/benefit analysis and identification of who loses and wins. If available time and resources are limited, GAM is most appropriate for a simple but effective means of displaying preferences. It is also important to identify the significant features of each alternative and to evaluate their impact with respect to the task being performed.

Tolerance of approximate data. A limited number of persons and a limited amount of means is usually available for the environmental decision-making, because the levels of investment in environmental matters are frequently lower than the levels of concern that public and private organizations express. This far-from-perfection situation requires to be able to achieve results from data not very accurately obtained. Thus, approximate reasoning models should be included to manage this uncertainty.

Now that we have introduced the features of EDSSs and the basic terminology of the field, we present a review of examples of this area of research.

4.4 Examples of EDSSs

In this section we describe some examples of EDSSs, including general applications in various environmental domains and specific implementations for WWTPs (Cortés *et al.* 2000).

- StormCast (Hartvigsen and Johansen 1990) is a distributed-AI application where typical real-time response times are not needed. It has been built to support storm forecasts over the Scandinavian peninsula and it may be described as a set of co-operating agents, which continuously collect and process weather data from a fixed geographical area. At each location, there is an expert module (a RBES) responsible for the prediction of severe storms. This forecasting is based on the results achieved by the monitoring agents in their own areas (problem solving at a local level).

- FRAME (Calori *et al.* 1994) is a knowledge-based tool to model air-pollution. The system is based on a rule-based expert system for the explanation and help phases. It includes an algorithm to determine user's expertise and to give selective access to information accordingly. FRAME's model-base is a *frame representation system* (FRS) that contains all the information and meta-information about the models. The selection of a suitable model usually depends both on aspects connected to the physics of the problem to be simulated and on the available resources.

- DCHEM (Distributed Chemical Emergencies Manager) (Avouris 1995) is an EDSS which supports decision making for the management of a specific class of environmental emergencies: accidents involving electrical equipment containing toxic chemicals. It is one of the first systems that uses distributed agent-technology and includes negotiation protocols in the problem-solving process.

- WATERSHEDSS (Water, Soil and Hydro-Environmental Decision Support System) (http://h2osparc.wq.ncsu.edu), developed by North Carolina State University's Water Quality Group in 1995, has as primary objectives to: (1) transfer water quality and land treatment information to watershed managers in order to assist them in making appropriate land management and land treatment decisions to achieve water quality goals and (2) assess and evaluate sources, impacts, and potential management options for control of non-point source pollution in a watershed based on user-supplied information and decisions.

- The Water Resources Institute's Information Service Center[5] has developed a series of DSSs in which accurate geographic and natural resources related information is presented in a form that can be easily used by a municipality in the development and implementation of land use policy. The application of the DSSs gives local officials, planners and engineers the ability to examine the potential impact of various possible decisions. ArcView software by Environmental Systems Research Institute (http://www.esri.com/) is used in some DSS as the principal software to improve the efficiency in making land use planning and zoning decisions.

- DAI-DEPUR (Sànchez-Marrè *et al.* 1996) is a distributed, integrated, multi-level, agent-based architecture for WWTP supervision and management. It combines in a single framework several cognitive tasks and techniques, such as learning, reasoning, knowledge acquisition, distributed problem-solving, and different AI techniques, such as rule-based reasoning and case-based reasoning. Four levels are distinguished from the domain-model point of view (Steels 1990): data, knowledge, situations and plans. On the other hand, from the supervision-task point of view, five levels are considered: evaluation, diagnosis, supervision, actuation and learning. The various subsystems forming DAI-DEPUR's architecture (e.g., the supervisory, the CBR, the primary settler KBS and the biological reactor KBS subsystems) can be executed in parallel. Distribution criteria are based on spatial and semantic distances. It is believed that a *supervisory system* is more efficient than other kinds of distributed AI systems, such as *blackboards systems* and *contract nets*, to deal with WWTP's

[5]http:// www4.gvsu.edu/wri/iscind.html

usual abnormal situations, such as *storm*, *bulking* and *toxic load*, through predetermined plans and actuation. The DAI-DEPUR system was developed for the WWTP domain, but it can represent a general framework for complex-process supervision (Sànchez-Marrè *et al.* 1999).

- In 1996, and through 1999, EPA[6] funded the project *Advancement of EDSSs through high performance computing and communication*[7]. The goal of the this research was to overcome computational resource limitations by developing tools for use within a *high performance computing and communications* (HPCC) environment, bringing *decision support systems* (DSSs) closer to fulfill the decision making power of a true cognitive system. To meet this goal, there were four primary research objectives: 1) to explore further the role of various optimization techniques in a DSS framework for complex environmental problems, 2) to examine ways of making better use of existing and expected future computational power to increase performance, 3) to develop better DSS prototypes and 4) to evaluate each prototype's performance and user interface with respect to user needs.

- Tripel (Rickel and Porter 1997) is the implementation of a method for constructing a model appropriate for answering a prediction question (compositional modelling). The method is evaluated in the domain of plant physiology. Given a prediction question and some domain knowledge, Tripel builds the simplest differential-equation model that can adequately answer it and automatically passes the model to a simulator to generate the desired predictions. Tripel uses knowledge of the time scales on which processes operate to identify and ignore insignificant phenomena and choose quasi-static representations of fast phenomena. It also uses novel criteria and methods to choose a suitable

[6]http://es.epa.gov/

[7]http://www.epa.gov/HPCC/homep.html

system boundary, separating relevant subsystems from those that can be ignored. Because its methods are domain-independent, Tripel should be useful in many areas of science and engineering.

- Sazonova and Osipov (Sazonova and Osipov 1998) created an EDSS oriented to the evaluation and prediction of marine fish stocks and to the determination of fishing quotas. As knowledge sources the system uses direct observations and dependencies suggested by experts.

- The North Carolina Supercomputing Center's Environmental Programs group[8] designed a system called simply *Environmental Decision Support System* (Fine *et al.* 1998), which is a problem-solving environment that provides a modelling and analysis system, for environmental scientists, engineers, policy makers and educators, in the domain of air quality. This system's design-goals include allowing modelers and decision-makers to generate, incorporate and understand new information with minimal effort, providing flexibility to model diverse issues and scales and contributing to a community modelling and analysis system[9]. This system includes several components and tools for building air quality models: an air quality simulation platform, a model configuration manager, a package for analysis and visualization, an emissions processing package, a graphical computation manager, an experimental optimization-based strategy development tool. This project originated as part of a cooperative agreement with EPA's Office of Research and Development in order to help the agency develop the next-generation air quality modelling system. It has expanded via a number of prototyping projects and is moving toward encompassing other

[8] http://envpro.ncsc.org/

[9] A *community modelling and analysis system* (CMAS) is an approach to development, application and analysis that leverages the community's complementary talents and resources in order to set new standards for quality of science and reliability of application of environmental models. The resulting comprehensive system forms the foundation which the community, including governments, industry, academia and other stakeholders, uses in the examination of issues and the subsequent development of strategies that meet societal challenges of environmental protection.

aspects of environmental decision support.

- BIOMASS (Besòs Intelligent Operation and Management of Activated Sludge System) (Comas 2000) is a supervisory system applied to wastewater treatment plants, which integrates the capabilities of a RBES (which can reason and explain its reasoning and conclusions) with a DSS (which manages data and decisions-making). The core and main modules of the system have been developed in the proprietary G2 object-oriented shell, which makes difficult further research efforts by others, due to unjustifiable high costs. On the other hand, G2 is a user-friendly development environment and embodies the inference engine. Apart from the G2 core, which controls on-line and off-line data acquisition, database management, rule-based reasoning and case-based reasoning (with plain memory), BIOMASS includes SCADA and PLC networks with basic control algorithms. Knowledge representation is achieved through classes and rules of different kinds (if, when, whenever, unconditionally, for, initially). BIOMASS has an objects base that is hierarchically structured. Most common units and objects in general, which are present in WWTPs, are stored into the objects base as classes and subclasses. Instantiations of these classes can then be connected among them to build the scheme of a specific WWTP. In the characterization of a case-study WWTP, 300 classes have been defined, together with 1000 descriptors. From the point of view of implementation, knowledge is organized into G2 workspaces. The main ones are: general knowledge (RBES, CBRS, supervision rules), data, definitions, microbiological identification, WWTP top-level (plant diagrams, trend charts, descriptors' discretization).

Now that we have presented a review of examples of EDSSs in the field of environmental science, we describe in more detail how AI paradigms integrate in this class of systems to give rise to decisions.

4.5 AI-paradigm integration

Environmental management can be a daunting task because so many factors must be taken into account. Environmental decision-making involves understanding not only the immediate impact of human activity on the environment, but also issues regarding human health, economic costs, current and pending regulation, fairness, and sustainability. In principle, all of these interrelated factors have a bearing on any decision related to the environment. In an area as complex as the environment, the integration of various AI paradigms seems to be necessary and can help in many ways.

Broadly speaking there are three main domains in which AI-paradigm integration can make a real difference. One domain is the modelling of complex environmental processes. Examples are the modelling of air quality, and water quality and treatment. Models can range in scope and sophistication from simple formulas that can be evaluated on PCs to massive programs that run most effectively on state-of-the-art supercomputers. The second domain is information management. Integrating information from diverse sources is necessary to make informed decisions. Main sources of information range from field-monitored data to simulation results to documents on regulatory policy and often different sources of are best dealt with by different AI techniques. The third domain involves modelling the decision process itself and thereby providing the structure and support to enable policy makers to take timely, balanced decisions that are consistent with the available knowledge about the environment.

More specifically, EDSSs including AI-paradigm integration are mainly used in tasks such as diagnosis, planning and optimization.

In this section, an interdisciplinary way to integrate the different approaches previously presented is discussed (Ceccaroni *et al.* 2000b). Considering the case of wastewater, an integrated architecture should be able to handle incomplete, uncertain and approximate information and would have to include (Venkatasubramanian 1994):

- monitoring: the management of sensors and continuous-analysis equipment;
- domain modelling: an ontology for WWTP behavior;
- numerical control: a tool to deal with basic water-quality characteristics and cost-reduction;
- quantitative-information control: the management of data such as water inflow, pH, dissolved-oxygen concentration and sludge recirculation-flow;
- qualitative-information control: the management of microbiological information and of data such as water color, odor and appearance;
- expert knowledge: the experience of managers, biologists and operators of the plant;
- experiential knowledge: the specific knowledge supplied by previously solved problems.

Such an architecture not only has to include all these elements, but also has to integrate them efficiently in real time. And, often, to easily acquire domain knowledge and to learn from past experience can be problematic. Powerful knowledge-acquisition tools are needed in order to cope with problems on the border of the domain of competence, and periodic update and maintenance are necessary to avoid knowledge degradation over time. Theoretical contributions which support the combination of different kinds of cognitive processes, such as rule-based reasoning, case-based reasoning, learning, knowledge acquisition and problem solving, are the ones of Agre (1996), Plaza *et al.* (1993), Newell (1990) and van Lehn (1990).

An example of integration of the various kinds of knowledge to model and manage the WWTP domain is in Sànchez-Marrè *et al.* (1996). This paper specifically describes the cooperation[10] among:

- a classic control algorithm, which models part of the physical and chemical numerical-knowledge;
- a set of inference rules which models part of the data and the expert knowledge;
- a case-based reasoner which models the experiential knowledge.

Despite the knowledge-integration efforts of this EDSS, several problems remain unsolved and impasse situations exist. To improve such a system, the following issues have to be dealt with: modelling of microbiological information for a higher reliability, better definition of the domain, description of a precise common terminology. The research on these points, which will also improve the reusability of the system in question, is one of the goals of this thesis and will be described in the next part.

[10]Cooperation, here, means to get benefit from the advantages of each kind of knowledge manipulation, being able, in this way, to cope with typical shortcomings both of KBSs and automatic-control systems.

Part II

An ontology-based environmental-DSS

This part of the dissertation describes the tasks undertaken to meet the objectives of the thesis. In chapter 5, we describe the creation and the development of a prototype decision-support system (OntoWEDSS) for the wastewater environmental domain. The architecture of the system has a modular design, to improve understandability, reliability and above all modifiability. Three modules, covering rule-based reasoning*,* case-based reasoning *and* ontologies*, are built for the management of a complex environmental process. With respect to the rule system, it is designed to be implemented in* two separate layers*. A more general one, which can be reused across WWTPs, and a more specific one to be used only in a particular WWTP. In chapter 6, we introduce the ontology (WaWO) that is embedded in the OntoWEDSS living environmental decision-support system. The essential addition of an* ontology *in the OntoWEDSS system helps to model the wastewater treatment process, paying a special attention to the management of the qualitative knowledge, that is, the environmental information on micro-organism presence. As well as helping to model the domain, the ontology adds new capabilities to the decision support system and its integration contributes to the advancement in decision support systems research.*

Using Allegro Common LISP, we implemented our approach on a case study problem. The case study involves the development of ontology-based wastewater-treatment control strategies, based on the process model explored by Comas (2000) and the reasoning model analyzed by Sànchez-Marrè (1995). This thesis is built on top of these two previous theses and extends their research towards the solution of several problems related to the wastewater domain.

Chapter 5

OntoWEDSS

Most of our predictions are based on very linear thinking.

That's why they will most likely be wrong.

Vinod Khosla

One of the goals of our research is to explore how *rule-based reasoning* (RBR), *case-based reasoning* (CBR) and *ontologies* can be applied to make the use of DSSs more reliable and practical for the management of complex environmental problems. This goal implies a focus is in the areas of: 1) solving complex environmental problems using ontologies in a distributed computing environment, 2) improving supervisory systems for WWTPs, 3) generating different alternative strategies to solve problems, and 4) identifying features and prototyping a potential architecture for environmental decision support.

In this chapter we will deal with the major efforts accomplished in these areas:

1. previously developed DSSs and supervisory systems in the domain of interest: DAI-DEPUR and BIOMASS (Sànchez-Marrè 1995; Comas 2000);

2. a prototype DSS for wastewater management (DAI-DEPUR+) based on previously developed DSSs;

3. the design, development and implementation of a new prototype (OntoWEDSS: this thesis).

An initial prototype, called DAI-DEPUR, was developed by Sànchez-Marrè (1995). The features included by this system were described in §4.4. The DAI-DEPUR+ system (Ceccaroni 2000) derives from the DAI-DEPUR system; it is its direct evolution through the addition of the WaWO ontology, object of chapter §6.

Now, with the proposal of an architecture for a new DSS prototype, the ontology is not just added, but embedded and integrated into a living environmental decision-support system for wastewater treatment plants. The main characteristic of this new system, called OntoWEDSS, is the integration of several AI techniques (including an ontology for the representation of the wastewater treatment process). Integration efforts such as the one of this thesis can be probably classified as proto-integration. Certainly an integrated architecture cannot be just a *collage* in which disparate paradigms are linked together with some sort of conceptual tape and we think this is not the case of OntoWEDSS. At the same time, nevertheless, even a *collage* can be a good engineering ground and a place to stand while considering the strengths and weaknesses of forthcoming better built architectures.

The OntoWEDSS system is built to manage specific WWTPs, but the ontology-underpinned representation of the domain will ease its *portability* towards other WWTPs and other domains. The OntoWEDSS system is constantly under development in relation with the research of the Knowledge Engineering and Machine Learning (KEML) group at UPC. The OntoWEDSS system aims to go a step further in completing the comprehension of WWTP-micro-organisms, through the use of the ontology and exploiting the data on activated sludge.

In this chapter we first clarify the knowledge-acquisition process used and then explain the architecture of OntoWEDSS and its 3 layers: perception, diagnosis and decision support. In next chapter, we will focus on the WaWO ontology.

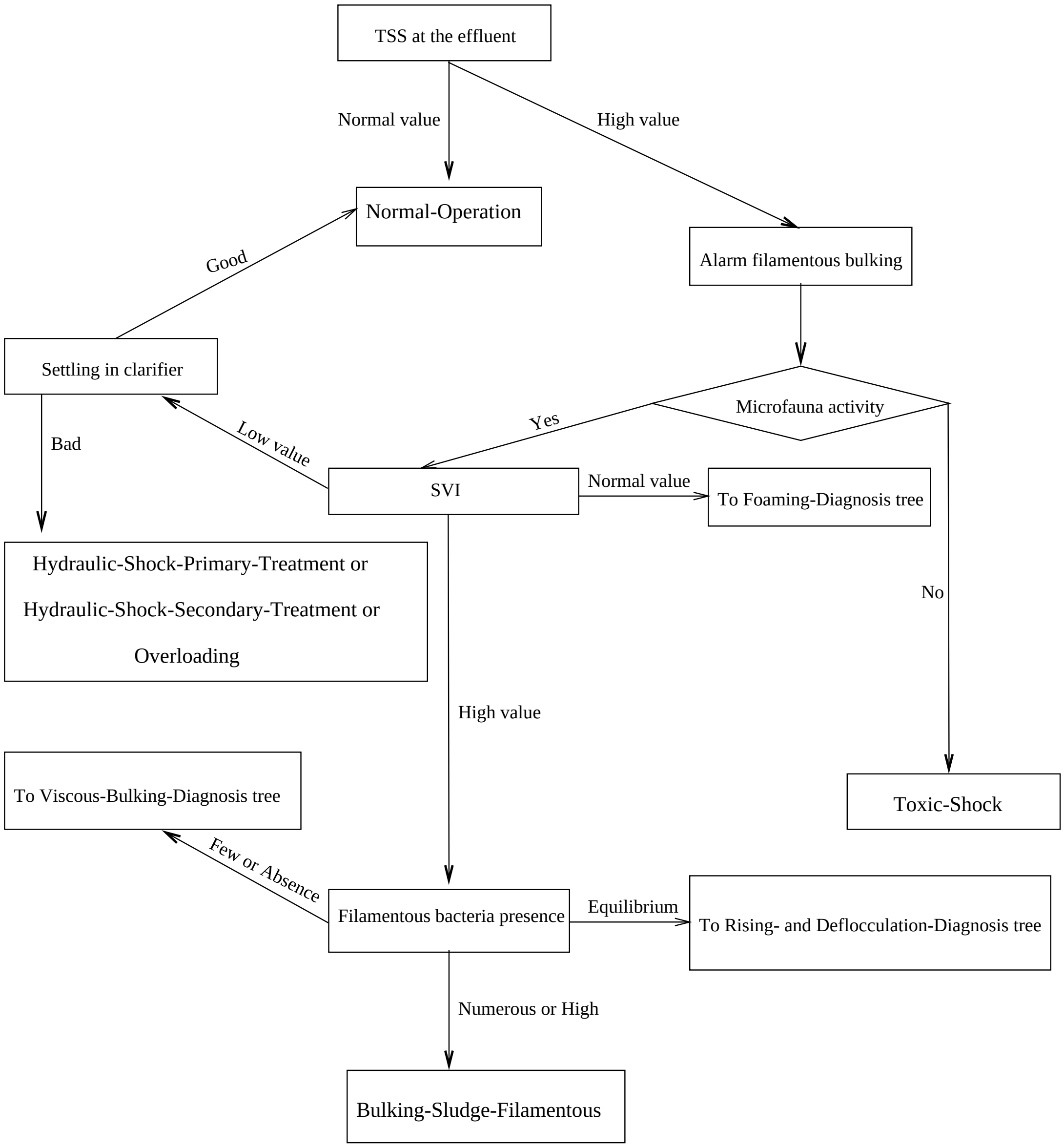

Figure 5.1: Main decision tree for filamentous-bulking problems (simplified).

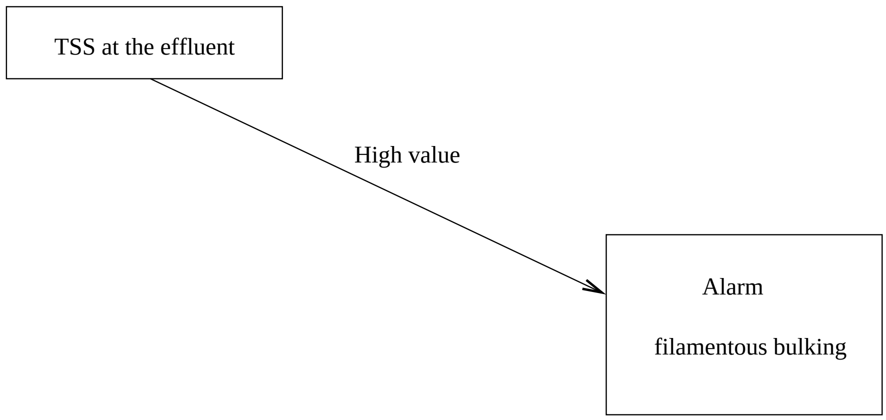

Figure 5.2: Part of a decision tree.

5.1 Knowledge acquisition

Of the three systems that we use, the expert system and the ontology require a knowledge acquisition process, in order to function. Sources of information for the KBs are: documents (existing literature about the WWTP domain) and experts (experience about the process). In this thesis we used manual knowledge acquisition methods (literature review and interviews) and reused a KB about WWTP management under the form of *decision trees* (Comas 2000). All symptoms, facts, procedures and relations used in diagnosis and decision support can be graphically represented with decision trees. They represent expert's procedural knowledge and decision-making behavior. These trees correspond to causal paths of interactions from symptoms to problems, using nodes interconnected by arches. Each node refers to a descriptor[1] or a test about a descriptor or an action, whereas each arch corresponds to a possible value for that descriptor or that test. There are three kinds of decision trees and they are used for the following tasks: diagnosis, cause identification and action strategy. In a diagnosis decision-tree (see Figure §5.1), leaf nodes represent a subclass of the WaWO's class *WWTP-Operational-State* (see Table §5.3). The translation of the knowledge contained in decision trees into *rules*

[1]See § 5.3.2 on page 114

is direct. For example, the arch and two nodes of Figure §5.2 identify the rule *IF TSS at the effluent has a high value THEN an alarm for filamentous bulking should be started*.

The decision trees we used are not active objects. The possibility to change cuts (between modalities of values) exists. For example, we can decide that the cut between *high value* and *normal value* for TSS is 10 instead of 20 mg/L. But, for the trees to be reused in other WWTPs, a practical way to change the descriptors in the nodes (or to change the destination of the arches) is needed. Otherwise the rule system reflecting the decision trees is too static for adaptation.

5.2 Architecture

The architecture of the system has a modular design, to improve understandability, reliability and above all modifiability. Modifiability is an important quality for software systems, because a large part of the costs associated with these systems is spent on modifications. The effort, and therefore cost, that is required for these modifications is largely determined by a system's software architecture. OntoWEDSS's architecture basically follows a standard vertical decomposition approach[2]: a division is made into many specialized subsystems, such as perception, diagnosis, modelling, planning, execution and effector-control modules.

Figure §5.3 contains a block diagram of the top-level decomposition of this architecture. The system receives raw data from on-line sensors and the laboratory, goes through a reasoning process and sends commands, via the action component, to the on-line sensors and effectors. This differs from many other systems, in which the control of on-line sensors is responsibility of the perception component.

Excepting cases of failure, there is a continuous sensory data stream from all on-line sensors, which goes directly into the perception component, along with the

[2]For the definition and application of horizontal and vertical decomposition, see (Brooks 1986) and (Kaelbling).

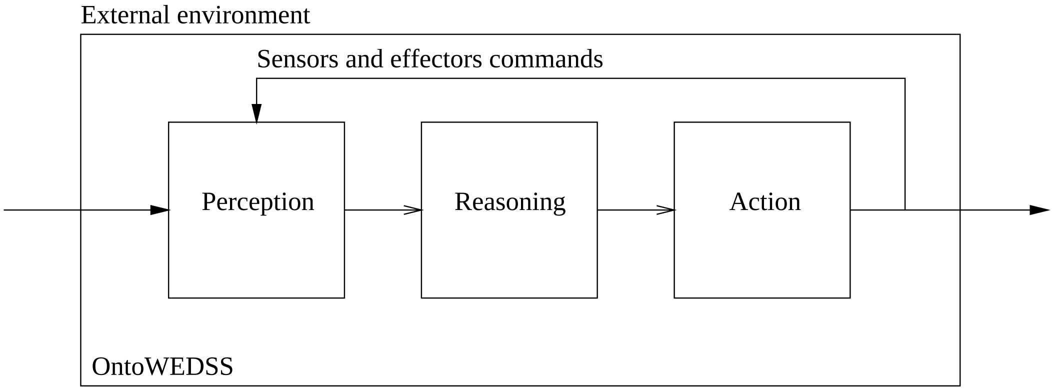

Figure 5.3: Top-level decomposition of OntoWEDSS.

results of laboratory analyses and the commands that were last sent to the effectors.

The detailed architecture of OntoWEDSS is schematized in Figure §5.4 and its action model is the following one:

- perception: data gathering and knowledge acquisition;
- diagnosis: reasoning and learning;
- decision support: prediction, evaluation of alternative scenarios, advising, actuation and supervision.

5.3 Perception layer

The OntoWEDSS system operates in a domain which physically consists of a wastewater treatment plant. In particular, all the physical, chemical and biological measurements are gathered in treatment plants located in Catalonia. Some descriptors are measured on-line by sensors, other ones are measured off-line, while other ones are calculated.

5.3.1 Awareness

The time scales of the treatment processes are long, so that the perception and the supervision decisions easily fit between sampling cycles. In a WWTP, sampling

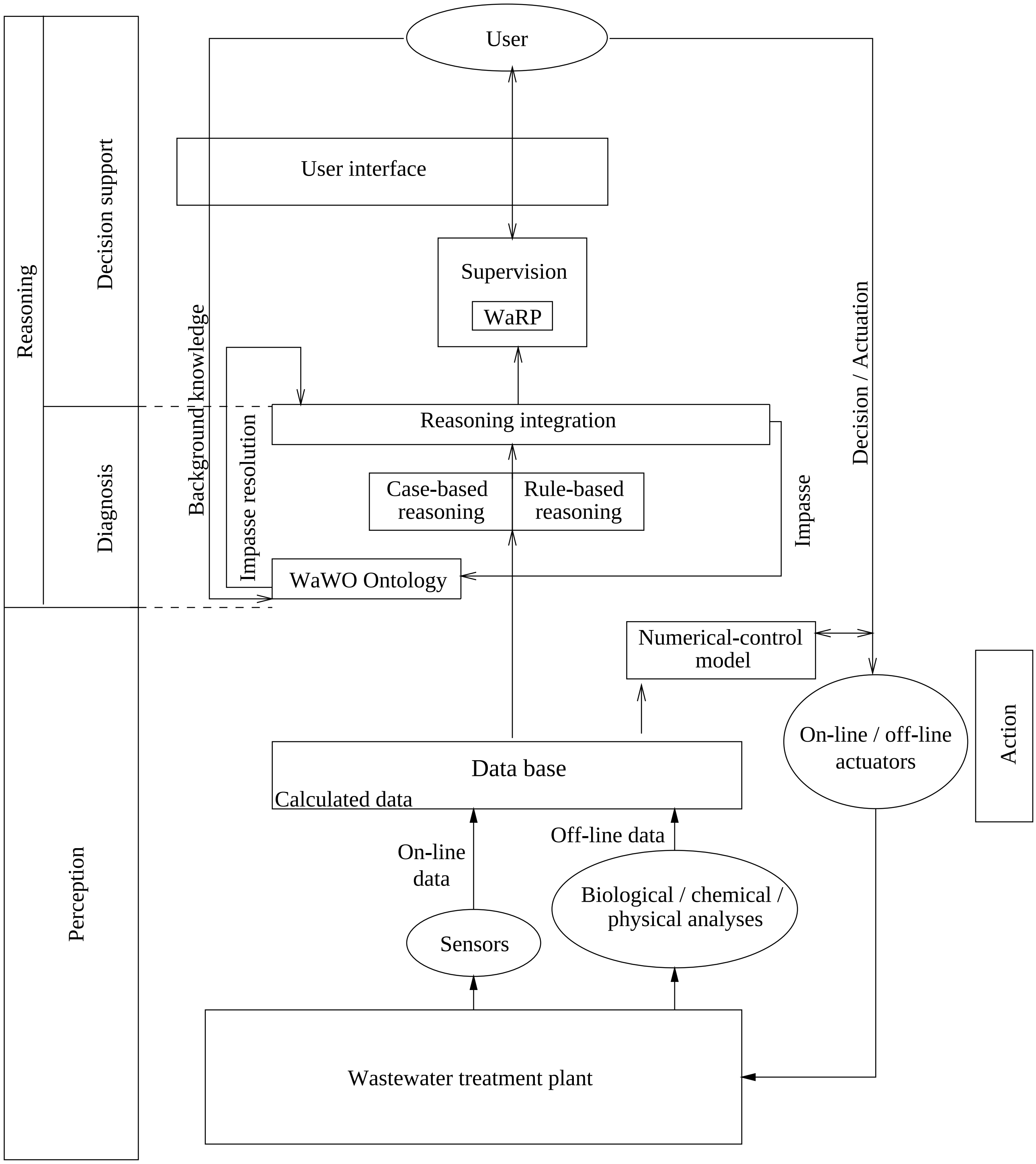

Figure 5.4: The OntoWEDSS architecture.

intervals of descriptors range from a few seconds to a few days. Our approach to the *temporal integration* of a number of processes that work at different rates is to define two constant cycle times for the entire system. This units of time are equal to 1 hour (for emergency detection) and 1 day (for action-strategy determination), and at each tick of theses clocks the inputs are read or calculated, some computation is done (e.g., means of accumulated data are calculated) and the outputs are set (by the action component of OntoWEDSS). If a process, such as a laboratory analysis, cannot be completed (or even cannot be started) by the tick of the time, either because its scheduling is non-constant or because its sampling interval is longer than the cycle's time or because there is a failure, its outputs are inferred, if possible, in an alternative way (often just reproducing the outputs of the previous cycle) and its execution is re-planned by the system for the following cycle.

Table 5.1: Descriptors of WWTPs as they appear in the WaWO ontology. (AT = aeration tank; S = settler; P = primary-treatment's effluent; I = inflow; E = WWTP's effluent)

Class	**Descriptor**	**Sampling location**
Off-line	*Qualitative descriptors*	
	Appearance-Floc	AT and S
	Appearance-Surface	”
	Biomass-Color	”
	Foam-Presence	”
	Water-Odor	”
	Water-Quality	”
	Biodiversity-Of-Ciliates	AT
	Biodiversity-Of-Filamentous-Bacteria	”
	Biodiversity-Of-Microfauna	”
	Dominant-Filamentous-Bacteria	”
	Flocs-Morphology	”
	Overall-Evaluation-Of-Floc-Quality	”
	Microfauna-Amoebae	”
	Microfauna-Ciliates	”
	Microfauna-Filamentous-Bacteria	”
	Microfauna-Flagellates	”
	Microfauna-Metazoa	”
	Microfauna-Unidentified-Ciliates	”

Class	Descriptor	Sampling location
	Total-Filaments	”
	Quantitative descriptors	
	Ammonia	I, P and E
	Biochemical-Oxygen-Demand (BOD)	”
	Chemical-Oxygen-Demand (COD)	”
	Chlorine	”
	Conductivity	”
	Greases	”
	Inhibitors	”
	Metals	”
	N-Total (TN)	”
	Nitrate	”
	Nitrite	”
	Oils	”
	Phosphate	”
	Phosphorous	”
	Temperature	”
	Total-Kjeldahl-Nitrogen (TKN)	”
	Total-Suspended-Solids (TSS)	”
	Turbidity	”
	Mixed-Liquor-Suspended-Solids	AT and RAS
	Mixed-Liquor-Volatile-Suspended-Solids	”
	V30	”
On-line	Dissolved-Oxygen	AT
	pH	I and E
	Sludge-Flow-Rate	I, P, E, AT, RAS and WAS
	Water-Flow-Rate	”
Calculated	%-BOD-Removal	- - -
	%-COD-Removal	- - -
	%-TSS-Removal	- - -
	Food-To-Micro-organism-Ratio	- - -
	Hydraulic-Residence-Time	- - -
	Sludge-Residence-Time	- - -
	Sludge-Volumetric-Index (SVI)	- - -
	Relative abundance of -	- - -
	- microbiological species	- - -
	Predominant filamentous organism	- - -
	Predominant protozoan	- - -

Many current DSSs simply *close their eyes* while a time-consuming subsystem, such as a planner or a reasoner, is invoked and the penalty for such unawareness is that perceptual inputs are either lost or stacked up for later processing. This is not the case in OntoWEDSS as the WWTP environment is evolving slowly compared to the speed of the reasoning of the decision support system: even if a WWTP is a truly dynamic domain, it never changes to such extent that the results of relatively long calculation would no longer be useful. If something that requires *immediate* action happens and this something is captured by on-line descriptors (on-line sensor-source descriptors in Table 5.1), OntoWEDSS is aware of it. Otherwise, it will be aware of it after the following reasoning-cycle (see §**??**).

5.3.2 Descriptors and data processing

In the general case, data are collected in various ways, classified and stored into a database. In Table B.1, a general data schema for WWTPs is shown, with sample values from the WWTP of Granollers, Catalonia. Each record in the database represent a complete day. For each day, about 130 descriptors[3] potentially exist[4], the names of which are listed in Table B.1. Three types of descriptors can be distinguished from the source point of view: on-line data coming from sensors; analytical, off-line data, both quantitative and qualitative; and calculated descriptors (Comas *et al.* 1999).

On-line source. On-line sensor data include diverse digital signals on the state of the mechanical equipment of the WWTP (e.g., pump's engine on/off, blower functioning at high/low velocity, electro-valve open/closed, floodgate open/closed, automatic grids' engine on/off). Apart from these mechanical equipment signals, on-line sensor data include many analogical signals, which are more interesting in view of the supervision task: flow rates (at inflow, primary effluent, plant effluent, by-pass, RAS, WAS, internal recycle and aeration) and physical descriptors, such as pH and dissolved oxygen concentration at each compartment of the aeration tank. On-line sensor data acquisition is accomplished by the PLCs network of the WWTP.

[3]Terminological clarification: in this thesis, with the term *descriptor*, we refer also to the following terms, used in the literature: attribute, character, characteristic, facet, feature, field, property, variable.

[4]Digital signals from some of the on-line sensors, being too dependent on the specific configuration of a WWTP, are not included in the list.

The set of PLCs is connected to a master-PLC, which transmits the whole on-line sensor data-set to the PC where the EDSS resides.

Off-line source. Analytical, off-line data can be quantitative and qualitative. Quantitative descriptors are provided by analyses of samples collected daily at various locations of the WWTP, while qualitative descriptors are supplied by daily *in situ* macroscopic observations or by laboratory microscopic examinations. A few quantitative descriptors (e.g., the conductivity in some WWTPs) are measured by mobile, not-on-line sensors. *In situ* observations are recorded through the control of WWTP's performance, quality of biomass and settling characteristics; they are recorded in the laboratory database once a day and from there transmitted to the PC where the EDSS resides. Microbiological determinations are generally carried out once or twice a week and they are not always recorded regularly (at constant intervals of time) in the laboratory database; therefore they are introduced manually to the PC where the EDSS resides, through a user-friendly interface.

Calculated descriptors. Combinations of quantitative data allow calculating more general process descriptors, such as sludge residence time, sludge volumetric index, F/M ratio, and percentages of COD, BOD and TSS removal by primary, secondary and overall treatment. The relative abundance [none, few, some, common, abundant, excessive] of the various microbiological species and the predominant filamentous organisms and protozoan is also calculated.

From an environmental point of view, data can be classified as physical, chemical and microbiological descriptors.

Physical and chemical descriptors. Among the available physical and chemical descriptors, the most relevant ones are selected on the basis of human experience, tradition and utility measures, although automatic methods exist that can make this selection. Chosen descriptors are then used by the OntoWEDSS system. The modelling and application of these descriptors, both in chemical engineering and AI systems, are well documented in bibliography (see §3.2.2).

Microbiological descriptors. The modelling of microbiological descriptors exists in the scope of biological disciplines, but it has not yet been integrated into a decision support system dedicated to environmental issues, such as the OntoWEDSS system. In this section we describe the methodology followed in the knowledge acquisition related to OntoWEDSS (Comas *et al.* 1999).

In a WWTP, the identification of the micro-organisms of the activated sludge is generally carried out at the laboratories of the plant and generates qualitative off-line data (e.g., presence of *Paramecia* species or diversity of *Ciliate*)[5]. A module to support the identification of microfauna, and in particular of filamentous organisms, by non-expert operators was developed (Comas 2000) on the basis of Jenkins *et al.* (1993) and Madoni (1994). When a micro-organism is identified, this microbiological-support module also gives the user information about the significance of its presence.

In the first phase of the development of the EDSS, after the identification of WWTP's micro-organisms, a comparative study of the communities of different treatment-plants was accomplished, to understand which is the influence of biological variability at a geographical level. A set of microbiological descriptors was then selected to be used by the system. For the maintenance of a high performance throughout the domain (the different WWTPs), this descriptors set needs to be broad enough to obtain a representational data-base with a relatively ample number of instances. Referring to portability, the descriptors available only in the minority of

[5]Using an automatic quantitative analysis of digital images, for micro-organism recognition and counting, is a possibility for the future.

the treatment plants are not very useful in the development of the main knowledge-bases of the system, but they can be used as specific-domain knowledge for specially developed modules. Finally, missing and incomplete information does not represent a problem in principle, but only a factor of increasing uncertainty.

Data about descriptors require to be validated, integrated into one uniform time-scale (see §5.3.1) and discretized before being used for diagnosis. We need to move from *raw* data to *interpreted* data.

Data validation. A checking procedure for the existence of the data and for correctness of the data values is carried out. This procedure can be summarized by the following sub-tasks:

- Detecting whether all data have been measured and whether they exist in the database in a machine-readable form.

- Monitoring the correctness of the values of the data (i.e., negative or out-of-range values for chemical analyses of wastewater and sludge; e.g., in a given plant, the inflow rate cannot be higher than 35,000 m^3/day).

- Updating the database. All the descriptors have a sampling rate (e.g., on-line sensor data are updated every 15 s, while analytical ones are imported once a day) and a validity time interval for their values (e.g., 36 hours for inflow-COD and 3 days for dominant-filamentous-organism). Once this interval has expired, the stored value of a descriptor is considered meaningless and a new value has to be acquired by the system or introduced by the user.

Discretization. Discretization is important in preprocessing of data. Any discretization process is defined by a set of cuts over domains of descriptors. In our study, quantitative descriptors are in general discretized using three modalities (two cuts): low, normal and high values (see example in Table § 5.2).

Table 5.2: Discretization of relevant descriptors of Granollers WWTP.

Descriptor	**Default weight**		**Modalities**	
		Low	Normal	High
Water-Flow-Rate-I	0.6	$\leq$ 19,000	19,000-24,000	24,000-60,000
COD-I	0.8	$\leq$ 650	650-850	850-3,000
TSS-I	0.7	$\leq$ 230	230-330	330-750
TKN	0.9	$\leq$ 60	60-90	90-200
COD-P	0.9	$\leq$ 525	525-720	720-1,500
TSS-P	0.7	$\leq$ 40	40-120	120-500
COD-E	0.8	$\leq$ 110	110-140	140-465
TSS-E	0.8	$\leq$ 10	10-20	20-95
TN-E	0.9	$\leq$ 60	60-90	90-200
MLSS-AT	0.7	$\leq$ 1,000	1,000-2,000	2,000-5,000
SVI-AT	0.8	$\leq$ 50	50-100	100-1,000
SRT-AT	0.7	$\leq$ 5	5-8	8-15
F-M-AT	0.8	$\leq$ 0.25	0.25-0.35	0.35-1.5
Filam-Dominant-AT	0.8	Microthrix	Gordona	none

When qualitative descriptors are used, the introduction of *fuzzy sets* is in general the best option, even if it is computationally more demanding. Specially in the case of general decision-trees (see §5.4.1), agreements about cuts among experts are not always possible and the use of fuzzy logic could possibly lead to better results and to an easier reuse of the knowledge base.

5.4 Diagnosis layer

Once all data have been interpreted, the use of diagnostic tools begins.

Diagnosis is a basic process for decision making in wastewater treatment. It is built on two different reasoning models: one based on general heuristic knowledge coming from literature and experts (RBES) and one based on specific experience accumulated through years of operation in a particular facility (CBRS). The integration of these two models permits to obtain very good results in wastewater treatment management. In OntoWEDSS, the result of diagnosis is a class of the WaWO ontology and specifically a subclass of *WWTP-Operational-State*. Table §5.3

shows the classification of WWTP's operational states and corresponds to WaWO's class *WWTP-Operational-State*.

Table 5.3: Classification of WWTP's operational states (more general classes on the left side).

	Top level	
WWTP-Operational- -State	Atypical-Situation	Electrical-Blackout
	Mechanical-Problem	
	Typical-Situation	Normal-Operation
		Operational-Problem
	Middle levels	
Operational-Problem	State-Abnormal- -Primary-Treatment	High-Solids-Loading
		Hydraulic-Shock-Primary-Treatment
		Inadequate-Sludge-Purge
		Low-Efficiency-Of-Grit-Removal
		Mechanical-Problem-General
		Old-Sludge
		Primary-Clarifier-Problem
		Pumps-Or-Pipes-Blocked
		Septic-Sludge
		Sludge-Removal-Systems-Break
		Too-High-Sludge-Density
	State-Abnormal- - Secondary-Treatment	Biological-Origin
		Non-Biological-Origin
	Lower levels	
Biological-Origin	Bulking-Sludge	Bulking-Sludge-Filamentous
		Bulking-Sludge-Slime-Viscous
	Deflocculation	Disperse-Growth
		Pinpoint
	Rising-Sludge	
	Dispersed-Growth	
	Foaming-Sludge	Foaming-Sludge-Actinomycetes
		Foaming-Sludge-Microthrix
		Foaming-Sludge-Gordona
	Pin-Point-Floc	
	Rising-Sludge	
	Toxic-Shock	
Non-Biological- -Origin	Aeration-Problem	
	Hydraulic-Shock- -Secondary-Treatment	
	Imbalanced-Flow-Rate	
	Mechanical-Electrical-Problem	

Meteorological-Conditions-Adverse	Storm
Overloading	
Secondary-Clarifier-Problem	
Surfactant-Scum	
Under-loading	

5.4.1 Knowledge-based paradigm

In the OntoWEDSS system there are two KBSs: a rule-based expert system (RBES) and a case-based reasoning system (CBRS). These KBSs, together with several automatic-control algorithms:

- detect in which state the plant is, whether a *normal* state or one of the *standard abnormal* states, such as *Bulking-Sludge-Filamentous*, *Storm* or *Foaming-Sludge*;
- contribute to manage the general wastewater-treatment operation in these cases.

The circumstantial aspects of the OntoWEDSS system related to rule-based reasoning, case-based reasoning and their diagnostic potential are described in §7. Here, we characterize additional facets of the two KBSs of OntoWEDSS which have not been dealt with in §4.1 and §4.2, being them specific of OntoWEDSS's design.

Uncertainty treatment. The RBES uses a combination of the methods for uncertainty treatment of MYCIN (Shortliffe and Buchanan 1975) and of the possibilistic model. *MYCIN* is one of the best known expert systems and it uses *certainty factors* (CFs) as a way of modelling reasoning under uncertainty. A CF is a number between -1 and 1 that represents the change in our belief on some *hypothesis* (H). A positive number means an increase in the belief and a negative number the contrary. A value of 0 means that there is no change in our belief on the hypothesis. In this work we are particularly interested in the propagation of uncertainty for the combination of rules, e.g., given E1 → H and E2 → H together with their respective CFs, we are interested on the CF of H given that E1 and E2 are true. And MYCIN introduced methods for calculating this propagation of uncertainty. The *possibility theory* is being developed as an alternative to traditional information theory and is logically independent of the probability theory, even if they are related. Traditionally, mathematical possibilistic semantics has been based strictly on fuzzy sets and their interpretation in the context of psychological uncertainty and subjective evaluations, but research exists to extend interpretations and applications of the possibility theory beyond those of fuzzy sets; in particular, to develop a natural semantics of possibility for the purposes of qualitative modelling of complex physical systems (Joslyn 1994). In this thesis we consider the conventional possibilistic model based on fuzzy logic. *Fuzzy logic* is a superset of Boolean logic, that has been extended to handle the concept of partial truth: truth values between *completely true* and *completely false*. It was introduced by Lotfi Zadeh of UC/Berkeley in the 1960's as a means to model the uncertainty of natural language. Just as there is a strong relationship between Boolean logic and the concept of a subset, there is a similar strong relationship between fuzzy logic and fuzzy subset theory. In classic set theory, a subset U of a set S can be defined as a mapping from the elements of S to the elements of the set 0, 1, U: S → {0, 1}. This mapping may be represented as a set of ordered pairs, with exactly one ordered pair present for each element of S. The first element of the ordered pair is an element of the set S, and the second element is an element of the set 0, 1. The value zero is used to represent non-membership, and the value one is used to represent membership. The truth or falsity of the statement *x is in U* is determined by finding the ordered pair whose first element is *x*. The statement is true if the second element of the ordered pair is 1, and the statement is false if it is 0. Similarly, a fuzzy subset F of a set S can be defined as a set of ordered pairs, each with the first element from S, and the second element from the interval [0,1], with exactly one ordered pair present for each element of S. This defines a mapping between elements of the set S and values in the interval [0,1]. The value *zero* is used to represent complete non-membership, the value *one* is used to represent complete membership, and values in between are used to represent intermediate *degrees of membership*. The set S is referred to as the *universe of discourse* for the fuzzy subset F. Frequently, the mapping is described as a function, the membership function of F. The degree to which the statement *x is in F* is true is determined by finding the ordered pair whose first element is x. The *degree of truth* of the statement is the second element of the ordered pair. In practice, the expressions *membership function* and *fuzzy subset* can be used interchangeably. Here is an example. Let us talk about wastewater-sample and hydraulic-load. In this case the set S (the universe of discourse) is the set of wastewater-samples. Let us define a fuzzy subset *Hydraulic-Overloading*, which will answer the question "to what degree does a wastewater-description x indicate an *Hydraulic-Overloading* state?". *Hydraulic-Overloading* can be described as a linguistic variable, which represents our cognitive category of hydraulic-load. To each wastewater-description in the universe of discourse, we have to assign a degree of membership in the fuzzy subset *Hydraulic-Overloading*. The easiest way to do this is with a membership function based on the *Water-Flow-Rate-I* wastewater-descriptor, whose graph looks like Figure 5.5.

```
Hydraulic-Overloading(d) =
  {0,   if Water-Flow-Rate-I(d) < 19,000,
   (Water-Flow-Rate-I(d)-19,000)/5,000,
        if 19,000 <= Water-Flow-Rate-I(d) <= 24,000,
   1,   if Water-Flow-Rate-I(d) > 24,000}
```

If *Water-Flow-Rate-I* on 29.10.1999 is 22,000 then *Hydraulic-Overloading* (29.10.1999) = 0.6. Membership functions used in most applications almost never have as simple a shape as *Hydraulic-Overloading*(d). Also, membership functions can be based on more than a single criterion. One could, for example, want to have the membership function for *Hydraulic-Overloading* depend on both *Water-Flow-Rate-I* on day d

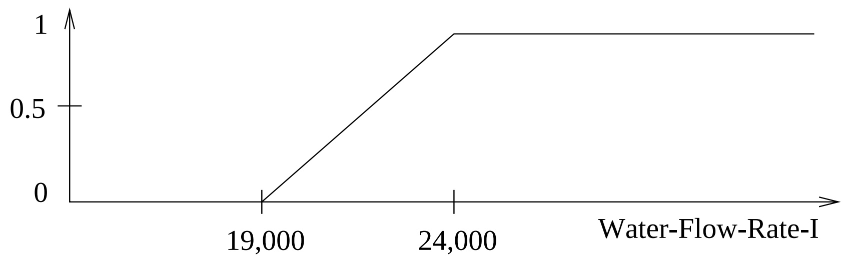

Figure 5.5: Fuzzy graph of a membership function.

and *Water-Flow-Rate-I* on day *d*-1. This is perfectly legitimate, and occasionally used in practice. It is referred to as a two-dimensional membership function, or a *fuzzy relation*. Now, how do we interpret a statement like "X is Low and Y is High or (not Z is Normal)"? The standard definitions in fuzzy logic are:

truth (not x) = 1.0 - truth (x)

truth (x and y) = minimum (truth(x), truth(y))

truth (x or y) = maximum (truth(x), truth(y))

This is not, however, the only possible interpretations of the logical operations. If you plug the values *zero* and *one* into the previous definitions, you get the same truth tables as you would expect from conventional Boolean logic. This is known as the *extension principle*, which states that the classic results of Boolean logic are recovered from fuzzy logic operations when all fuzzy membership grades are restricted to the traditional set $\{0, 1\}$. This effectively establishes fuzzy subsets and logic as a generalization of classic set theory and logic. In fact, by this reasoning all crisp (traditional) subsets *are* fuzzy subsets of this very special type; and there is no conflict between fuzzy and crisp methods.

Lack of critical information. As seen in §5.1, the RBES is based on the translation of decision trees to rules. Apart from the problems described in §4.1.2, a complication related to decision trees is that there is no procedure defined for the case of lack of critical information across a tree. This problem is tackled by the addition of extra rules to the expert system and by the use of the CBRS and WaWO.

With respect to reusing the rule system, it is important to implement *two separate layers of decision trees.* A more general one, which can be reused across WWTPs, and a more specific one to be used only in a particular WWTP.

Case-library's structure. In CBRSs, the case-library's structure can be flat or hierarchical. In OntoWEDSS the structure is flat and sequential and is implemented in the form of a non ordered list. The reasons for this choice are: (1) a flat structure is the simplest and less costly way to introduce a new case; (2) it always retrieves the set of cases which best match the new case; and (3) the architecture of the CBRS is open and modular and to change the case-library's structure is not problematic, if needed. On the other hand, the disadvantage is that the retrieval time can be very long if the case library has a considerable size. In the design and implementation of the prototype we do not work with very large libraries and a flat structure is a reasonable option. If a situation in which a hierarchical structure is preferred is reached, then the case library can be organized using a lazy prioritized tree (Sànchez-Marrè *et al.* 1997), where each node corresponds to the evaluation of a descriptor and terminal nodes contain stored cases. In this way the retrieval time is reduced. To build the prioritized discrimination tree, discretized descriptors are used. Descriptors' weights (used in similarity assessment) and discriminant order (descriptor with order 1 is the one at the root of the discrimination tree) are subjective and are decided by experts after an iterative process of trial and error, which aims to balance retrieval-time and similarity optimization. In the case of the Granollers WWTP, the default weights of descriptors are in Table §5.2 and the discrimination order of some of them is: 1) TSS-E, 2) SVI, 3) COD-E, 4) COD-I, 5) TSS-I, 6) Water-Flow-Rate-I, 7) MLVSS-AT, 8)TKN, 9) TN-E. The weight of a descriptor is modified if, at the introduction of a new case, the descriptor has an abnormal value. In this case the operational state of the WWTP is not likely to be *Normal-Operation* and the relevance of the abnormal descriptor in the characterization of the actual state is considered higher.

This weight change is managed by rules like:

IF Water-Flow-Rate-I < 13,000 m^3/day **or** Water-Flow-Rate-I > 30,000 m^3/day

THEN weight(Water-Flow-Rate-I) = 10

Missing values. If the value of a descriptor is missing, the descriptor is not pondered in calculating the similarity between cases. As well as domain discretization-cuts, descriptors' weights can be modified by the user by means of the *application program interface* (API).

Similarity in CBR. The *similarity* (β) between cases is defined as complementary to their distance (D). It can be expressed as β = 1 - D and its range is [0,1].

Specific knowledge. The CBRS is often able to model also specific features and particular states of the treatment plant (*non-standard abnormal* states), and to learn from past situations occurring in the treatment plant itself. This would account for the potential difference in individual treatment-plants due to deviations in parameters such as inflow, meteorology, neighboring industries and local life-style (Sànchez-Marrè 1995; Sànchez-Marrè *et al.* 1996).

5.4.2 Ontology

An ontology (fully described in chapter 6) is integrated with the KBSs mentioned in §5.4.1. With this ontology it is possible to capture, understand and describe the knowledge about the whole physical, chemical and microbiological environment of a WWTP. The hierarchical structure and the axioms of the ontology can help to diagnose the situation in case of *impasse* of the other KBSs.

The ontology is normally static. It activates its inference mechanisms (axioms) only under specific requests from the diagnosis integrator (see §5.4.3). The result of the inference of the ontology is both:

- an answer about the diagnosis impasse (e.g.: 'We have a foaming situation' or 'I do not have information to solve the impasse');

- an explanation of the answer (e.g.: 'I received information related to the answer from the activation of the following axioms ...' or 'The answer was obtained searching the following classes ...').

The activation of the ontology always means that there exists an impasse in KBS diagnostic process. If the answer of the ontology to a given request is *I do not have information to solve the impasse*, then a primary alarm is activated.

5.4.3 Diagnosis integration

The rule-based expert system (RBES) and the case-based reasoning system (CBRS) work independently and they both produce as output a diagnostics on the state of the plant. This output is passed to the diagnosis integrator (*Reasoning integration* box in Figure § 5.4 on page 111). This integrator shows the two outputs (diagnostics together with an associated confidence value) to the user and then starts the reasoning schema detailed below (see also §7.4).

General integration schema. If the diagnostics of the two KB systems are the same, this result is communicated to the decision support layer. If the diagnostics exist and are different, the system prioritizes as follow (see §5.6):

- If the case similarity is higher than a predefined threshold, the case-based reasoner's diagnostics prevails.

- Otherwise, the rule-based expert system's diagnostics prevails.

In case of impasse (no diagnosis), OntoWEDSS turns first to the ontology and then, if it fails, to the plant manager, demanding an off-line diagnosis based on their microbiological deep knowledge. This external solution is always learned.

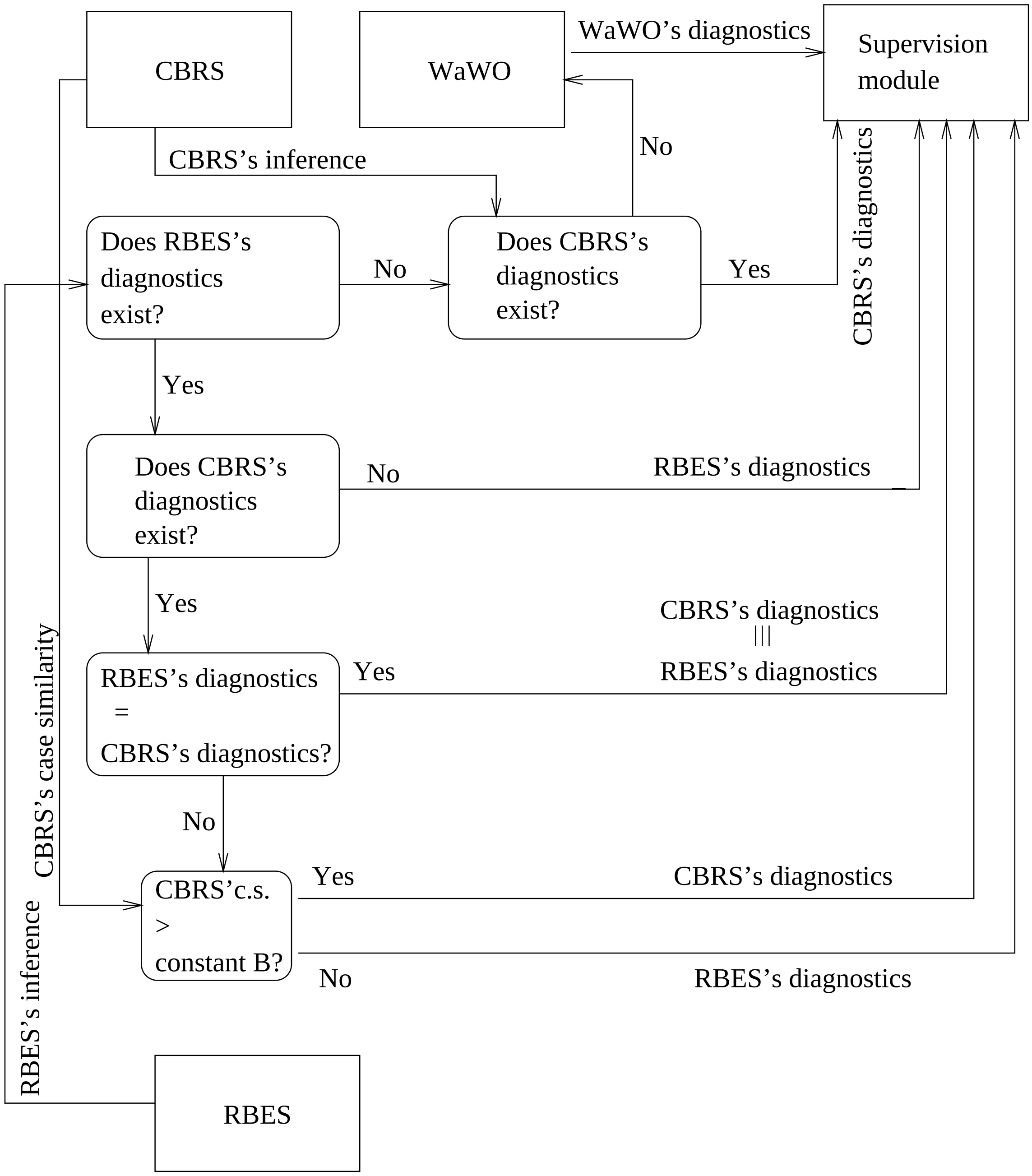

Figure 5.6: Data flow of general diagnosis-integration.

Detailed integration schema

```
∄ CBRS diagnostics and ∄ RBES diagnostics:
   impasse, the diagnosis integrator turns to the WaWO ontology.

∄ CBRS diagnostics and ∃ RBES diagnostics:

   RBES diagnostics-certainty ≥ α: RBES diagnostics passed
   to decision support layer.

   RBES diagnostics-certainty < α: impasse, the diagnosis
   integrator turns to the WaWO ontology.

∃ CBRS diagnostics and ∄ RBES diagnostics:

   CBRS case-similarity ≥ β: CBRS diagnostics passed to
   decision support layer.

   CBRS case-similarity < β: impasse, the diagnosis
   integrator turns to the WaWO ontology.

∃ CBRS diagnostics and ∃ RBES diagnostics:

   CBRS case-similarity ≥ β: CBRS diagnostics passed to
   decision support layer.

   CBRS case-similarity < β:

      RBES diagnostics-certainty ≥ α: RBES diagnostics
      passed to decision support layer.

      RBES diagnostics-certainty < α: impasse, the diagnosis
      integrator turns to the WaWO ontology.
```

Example We have a certain perception state A. State A is characterized by a strong presence of Gordona bacterium. The case-based reasoning system (CBRS) and the rule-based expert system (RBES) are activated. The Gordona bacterium has no direct relation in the knowledge base of the RBES to any state of the WWTP and the RBES finds no rules leading to a diagnostics starting from state A. Therefore there is no diagnosis output from the RBES. The CBRS finds a case similar to state A. The similarity value (see §5.4.1) is 0.1 and it is less than the minimum acceptable value β (e.g. β=0.6; this value can be decided and changed by the expert). Therefore there is no diagnosis output from the CBRS.

The diagnosis integrator acknowledges a case of missing diagnostics from the KBSs and send a request to the WaWO ontology with the description of state A. Even in the ontology the *Gordona* class has no link to any state of the WWTP, but its parent class (*Filamentous-Bacteria*) has a cause-effect link to the general state of the WWTP *Foaming-Sludge*. One reason for this could be that *Gordona* is not causing foaming, but another (not detected) bacterium of the same taxonomic class is. The output of WaWO is:

- Answer = 'We have a foaming situation.'
- Explanation = 'I received information related to the answer from the activation of the following relation (Gordona *is a* Filamentous-Bacteria) and the following relation (Filamentous-Bacteria *is cause of* Foaming-Sludge).'

The diagnosis integrator receives the diagnostics from WaWO and pass it to the decision support layer.

5.5 Decision-support layer

This layer exploits available data and information to provide active decision-support about key actuations in the WWTP. This layer includes the user interface and the supervision module.

Example When several factors may contribute to the need for the operator to perform a control function, decision support is provided. For instance, effluent turbidity from a process unit, such as an activated-sludge final sedimentation basin, may be caused by bulking sludge, rising-sludge in the final basin or an improper sludge inventory in the basin. The operator is warned of the problem by an automated routine of OntoWEDSS and the integrated diagnostics of the reasoners is presented to him. Sometimes, a course of action is proposed. The operator reviews the situation and determines the control function required.

In summary, the decision-support task consists of gathering the integrated diagnostics of reasoners and ontology, activating the WaRP planner (Ceccaroni and Robertson 2000) and selecting an actuation. User is also given the possibility to ask the system for explanation about the results of the rule-based and case-based reasoning.

In *normal situations*, a WWTP is guided by classic, automatic control. If the supervision module acknowledges an *abnormal situation*, it proposes to modify or deactivate the automatic control and presents an action strategy (based on general rules or on experience from past *abnormal situation* in the same plant) to get the process back under control. A set of supervision meta-rules are used to decide whether the actions are to be based upon expert or experiential knowledge. In general, the activating conditions of the action meta-rules are related to the result of the diagnosis process (often, action schemas are already included in the diagnosis result).

Being OntoWEDSS a decision support system and not an automatic direct manager, the diagnostics and the action schema are communicated to the user through the user interface. The user validates OntoWEDSS' output and then can instruct it to carry out suitable actions on the plant. It is always the manager of the WWTP to perform the final evaluation of the plans suggested by OntoWEDSS, deciding if the proposed strategy is really the best one to redirect the process towards the correct operation.

In *abnormal situations*, the WWTP's manager has usually two possibilities:

1. to deactivate the automatic control and implement the actions suggested by OntoWEDSS; in this case, after the execution of the actions planned or when a *normal situation* is established, the automatic control is re-enacted;

2. to maintain, even with modifications, the automatic control over the WWTP

and to carry out parallel strategies (not proposed by OntoWEDSS) to accelerate the return of the process to normality.

At present, on-line actuators that can carry out automatically the actions planned by OntoWEDSS are modelled, but are not used to modify any descriptor (e.g., the set-points of the numerical control module, sludge and water's flow rates, valve and pump's positions). Nevertheless, there is no technical problem in deploying automatic actuators. It is just a matter of time, agreements with wastewater utilities and compatibility with local legislation.

The possibility of developing a module for reactive planning in the decision-support layer of OntoWEDSS has been explored during the research period at the Edinburgh University (Ceccaroni and Robertson 2000). A working prototype has been implemented in Prolog.

5.6 Why OntoWEDSS is better than earlier systems?

The earlier systems of reference for OntoWEDSS are DAI-DEPUR (Sànchez-Marrè 1995) and BIOMASS (Comas 2000).

OntoWEDSS improves these systems in several ways, but mainly by the introduction of an ontology and the addition of a diagnosis integrator. The introduction of the WaWO ontology helps in improving the reliability of decision support, in reusing knowledge and in facilitating the portability of OntoWEDSS. Unlike DAI-DEPUR and BIOMASS, in OntoWEDSS a semantic integration (Stuckenschmidt 2000) of information exists. In fact, a problem that goes beyond syntactic integration (dealt with by BIOMASS and in part by DAI-DEPUR) is the mapping of semantics of terms from different information sources (such as different WWTPs), even when these terms have been expressed using the same syntactic structures. For instance, even when two applications use the same language as their interchange format, how can we be sure that the same words in their vocabularies mean the same things? The WaWO ontology, which we introduced, is an instrument to solve semantic problems of this kind. Being WaWO stored in the well known Ontolingua Server (§2.4.7), the search for it is easy and the knowledge-representation formalism is standard. Moreover WaWO can be translated into several implementation languages thanks to Ontolingua translators. The lexicon and semantics of WaWO are as much standard as possible, synonyms are shown in the documentation and there are no hidden assumptions. Solving part of the existing terminological confusion, OntoWEDSS matches more properly the domain needs.

Impasse situations in DAI-DEPUR and BIOMASS are solved by the new ontology. In case of impasse (no diagnostics), OntoWEDSS turns to WaWO demanding an off-line diagnosis based on its microbiological deep knowledge. The hierarchical structure and the axioms of WaWO are what help to diagnose the situation in the case of an impasse of the other KBSs, allowing reasoning on different levels of abstraction.

While in DAI-DEPUR there is no modelling of wastewater microbiology, in the OntoWEDSS system the microbiological component is modelled by the ontology, and this opens new possibilities of search and inference in the process of WWTP control.

An identification of most common micro-organisms and a comparative study of micro-organism communities of different treatment-plants have been carried out, to understand what can be the influence of biological variability at a geographical level. A set of microbiological descriptors have been selected to be used by WaWO, together with the standard physical and chemical ones. With WaWO it is then possible to capture, understand and describe the knowledge about the whole environment of a WWTP.

OntoWEDSS presents a novel integration between KBSs and ontologies in a real-world application. The integration happens mainly at the diagnosis level, where the results of rule-based and case-based reasoning systems are compared before passing a final decision to the decision support layer. A system of priority is established among the KBSs as well as the cases in which WaWO is called.

OntoWEDSS incorporates cause-effect reasoning, thanks to the implementation of a set of relations that will enable WaWO to automatically deduce the answer to many questions about the wastewater domain (see chapter 8). Through the definition of ontological relations we represented mainly two kinds of real-world cause-effect relations: association of micro-organisms to the problematic situations that they cause and association of the actual state of the plant to the actions that need to be performed in order to reach the normal state from that actual state.

Chapter 6

WaWO

> The value of a theory doesn't consist in being right or wrong: the important fact is to suggest good experiments and to permit to advance in the learning process.

6.1 Why an ontology for wastewater treatment?

The main reasons for the development of an ontology for wastewater treatment are described in this section:

Terminological confusion. In the domain of wastewater treatment, many different experts work together. Their disciplines of expertise are: AI, chemical engineering, chemistry, computer science, environmental engineering, microbiology. Each expert uses a particular vocabulary (a precise common terminology does not exist). There are no rules helping in the use of each term. Synonyms exist. Some term can be used in different disciplines with similar, but not identical, meanings (semantic differences appear using the same term in different disciplines).

All these reasons reflect the need for the creation of a unified, complete and consistent terminology, which can be used in different formal contexts and applications related to the wastewater treatment domain. The *WaWO ontology* is a practical way to achieve this goal.

Domain modelling. The WaWO ontology wants to model not only the terminology but also the entire domain of wastewater-treatment processes and management, with particular attention to the activity of the microbiological component. Moreover, WaWO encodes a deeper microbiological knowledge with respect to existing knowledge-based systems.

One of the paradigms used as basis for domain modelling is object orientation. A very popular description form for modelling in the ontology domain are frames. Frames are well suited for the modelling of control systems and physical-biological systems. Ontolingua and OIL existing proposals use frame-language terms to design ontologies. Object-oriented modelling can be used also to model axioms about wastewater treatment in general, causal relations and non-causal energy flows.

Reuse. The WaWO ontology is created to be easily reusable. We clearly identify the environmental processes involved and the development steps in the ontology construction, but we do not assess the cost-effectiveness of the reuse of this particular ontology. Reusing an ontology is far from being an automated process. It requires not only consideration of the ontology, but also of the tasks for which it is intended. Before being reused, WaWO could have to be translated and trustworthy, fully automatic translators are unlikely to be forthcoming in the foreseeable future (Uschold *et al.* 1998). Nevertheless, we think that in many cases knowledge reuse can be cost-effective, and that it would take significantly longer to design the knowledge content of this ontology from scratch in other applications. In some cases, knowledge engineers working in other WWTPs will only need to worry about creating the specialized knowledge and reasoners relative to specific tasks of their system. In general, this approach should facilitate building bigger and better systems cheaply. It should also lead to a greater diffusion of these systems.

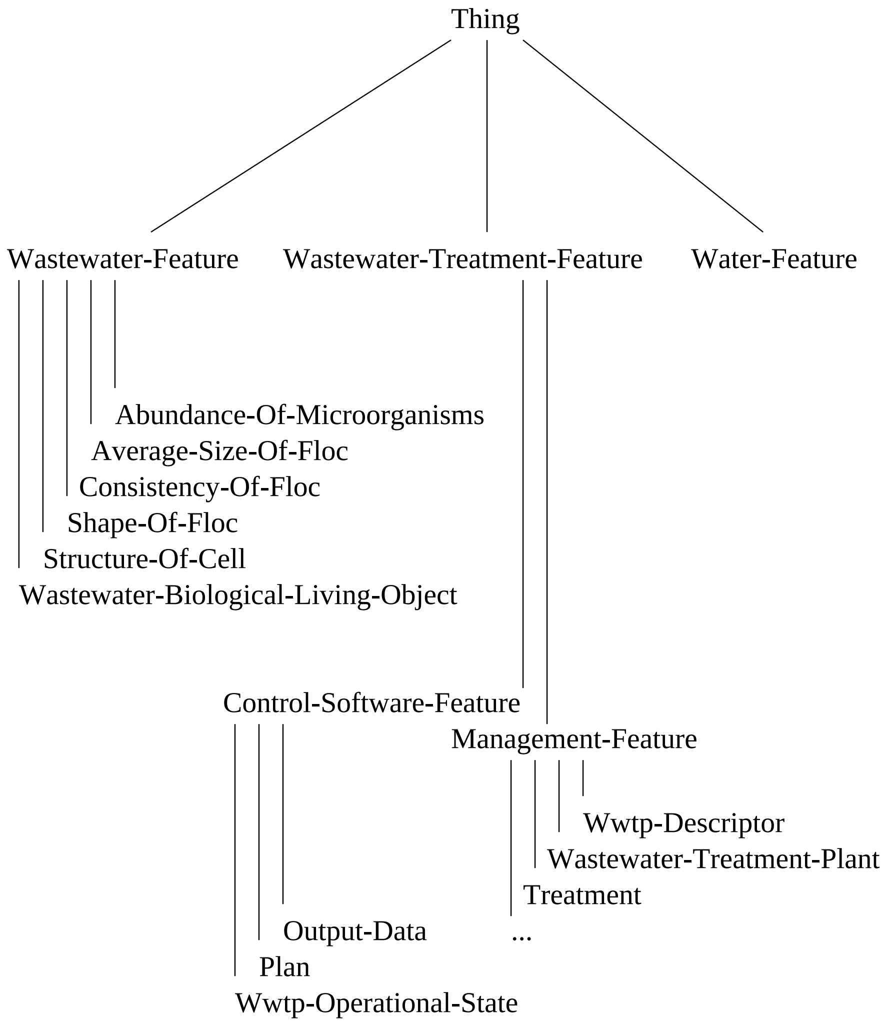

Figure 6.1: Top-level categories in the October 2001 version of the WaWO ontology.

6.2 Categories

A choice of general ontological categories and of a standard interchange language are the first steps in designing an ontology as well as any object-oriented system. In database theory the categories are usually called *domains*, in logic they are called *types* or *sorts*, in object-oriented systems they are called *classes* and in AI they are called *types* or *classes* (Sowa 2000). In WaWO description, we will use the terms *categories*, *types* and *classes* as synonyms. Whatever they are called, the selection of categories determines everything that can be represented in the OntoWEDSS application. Any incompleteness, distortion or restriction in the framework of categories inevitably limits the generality of the OntoWEDSS program, which uses those categories.

During the building the WaWO microworld, following the first design step, we identify the categories in our domain of discourse (the wastewater), according to current studies (e.g., Comas (2000)). At the highest abstraction level, we decide to classify these types in three top-level categories: *Wastewater-Treatment-Feature* (comprehending objects related to the WWTP management and AI control software), *Wastewater-Feature* (including also objects belonging to microbiological taxonomy) and *Water-Feature*. The complete indented list of WaWO class hierarchy is as follows:

```
Wastewater-Feature
   Abundance-Of-Micro-Organisms
   Average-Size-Of-Floc
   Consistency-Of-Floc
   Shape-Of-Floc
   Structure-Of-Cell
      Cell-Wall
      Chloroplasts
      Cytoplasmic-Membrane
      Endoplasmic-Reticulum
      Golgi-Apparatus
      Internal-Organelles
      Mitochondria
      Nucleus-Containing-A-Clear-Membrane-Surrounding-Dna
      Ribosomes
      Vacuoles
   Wastewater-Biological-Living-Object
      Bacterium
         Autotroph
         Heterotroph
         Filamentous-Bacterium
            Ramifications
            Motility
            Form-Of-Filament
            Localization
            Bacterial-Growth-Around-Filament
            Size-Of-Filament
```

- Size-Of-Cells
- Eukaryotic-Biological-Living-Object
  - Alga
  - Worm
  - WWTP-Fungus
- Wastewater-Virus

Wastewater-Treatment-Feature
- Wastewater-Treatment-Control-Software-Subdomain
  - Output-Data
    - Automatic-Control-Data
      - Sludge-Recirculation-External
      - Sludge-Recirculation-Internal
      - Sludge-Waste
      - Water-By-Pass-All-Treatment
      - Water-By-Pass-Secondary-Treatment
      - Aeration-Flow-Rate
      - Dosage-Of-Chemicals
        - Coagulants
        - Flocculants
        - Bactericides
        - Nutrients
      - Addition-Of-Alkalinity
      - Mechanical actuation
  - Plan
  - WWTP-Operational-State
    - Atypical-Situation
      - Electrical-Blackout
    - Mechanical-Problem
    - Typical-Situation
      - Normal-Operation
      - Operational-Problem
        - State-Abnormal-Primary-Treatment
          - High-Solids-Loading
          - Hydraulic-Shock-Primary-Treatment
          - Inadequate-Sludge-Purge
          - Low-Efficiency-Of-Grit-Removal
          - Mechanical-Problem-General
          - Old-Sludge
          - Primary-Clarifier-Problem
          - Pumps-Or-Pipes-Blocked
          - Septic-Sludge
          - Sludge-Removal-Systems-Break
          - Too-High-Sludge-Density
        - State-Abnormal-Secondary-Treatment
          - Biological-Origin

```
          Bulking-Sludge
            Bulking-Sludge-Filamentous
            Bulking-Sludge-Slime-Viscous
          Deflocculation
            Dispersed-Growth
            Pin-Point-Floc
          Foaming-Sludge
            Foaming-Sludge-Microthrix
            Foaming-Sludge-Gordona
          Rising-Sludge
          Toxic-Shock
        Non-Biological-Origin
          Aeration-Problem
          Hydraulic-Shock-Secondary-Treatment
          Imbalanced-Flow-Rate
          Mechanical-Electrical-Problem
          Meteorological-Conditions-Adverse
            Storm
          Overloading
          Secondary-Clarifier-Problem
          Surfactant-Scum
          Underloading
Wastewater-Treatment-Management-Subdomain
  Actuation
  Motor-Equipment
    Archimedes-Screw
    Blower
    Electrovalve
    Pump
    Turbine
      1-Speed
      2-Speeds
  Process-Equipment
    Belt-Filter-Press
    Bioreactor
      Activated-Sludge
        Carrousel
        Compartmentalized
        Complete-Mixed
      Fixed-Film
    Digester
    Grit-Chamber
    Screen
      Manual
      Narrow
```

```
      Wide
   Settler
      Clarifier
      Flotation-Unit
      Primary-Settler
      Thickener
   Sewer
   Tank
   Valve
      Valve-Automatic
      Valve-Manual
Sampling-Locations
Treatment
   Preliminary-Treatment
      Degreasing
      Sand-Removal
      Screening
   Primary-Treatment
      Chemical-Treatment
      Physical-Treatment
   Biological-Treatment
      Activated-Sludge-Treatment
   Thermal-Treatment
Wastewater
Wastewater-Treatment-Plant
   WWTP-With-Activated-Sludge
WWTP-Descriptor
   Calculated
      %-BOD-Removal
      %-COD-Removal
      %-SS-Removal
      Food-To-Micro-Organism-Ratio
         F-M-At
      Hydraulic-Residence-Time
      Sludge-Residence-Time
         SRT-At
      Sludge-Volumetric-Index
         SVI-At
   Day
   Off-Line
      Descriptor-Qualitative
         Macroscopic-Observation
            Appearance-Floc
            Appearance-Surface
            Biomass-Color
```

- Bubbles-Presence
- Floating-Sludge-Presence
- Foam-Presence
- Settling-Characteristics
- Structure-Of-Floc
- Water-Olor
- Water-Quality

Microscopic-Observation
- Biodiversity-Of-Ciliates
- Biodiversity-Of-Filamentous-Bacteria
- Biodiversity-Of-Microfauna
- Dominant-Filamentous-Bacteria
  - Filam-Dominant-At
- Floc-Characterization
  - Average-Floc-Size
  - Effect-Of-Filaments-On-Floc
  - Morphology
    - Morphology-Consistency
    - Morphology-Shape
    - Morphology-Structure
  - Overall-Evaluation-Of-Floc-Quality
- Microfauna
  - Filamentous-Bacteria
    - Beggiatoa
    - Haliscomenobacter-Hydrosis
    - Microthrix-Parvicella
    - Gordona
    - Nostocoida-Limicola-I
    - Nostocoida-Limicola-II
    - Nostocoida-Limicola-III
    - Thiothrix
    - Type-0041
    - Type-0092
    - Type-021-N
    - Type-0411
    - Type-0581
    - Type-0675
    - Type-0803
    - Type-0914
    - Type-0961
    - Type-1701
    - Type-1863
    - Zoogloea-Ramigera
    - S-Natans
    - H-Hydrossis

- Free-Bacteria
- Metazoa
  - Nematode
  - Rotifer
- WWTP-Protozoan
  - Amoebae
    - Nude-Amoebae
    - Testate-Amoebae
  - Ciliates
    - Bacteriophagic
      - Crawling
        - Acineria-Uncinata
        - Aspidisca-Cicada
        - Chilodonella
        - Euplotes-Spp
      - Free-Swimming
        - Colpidium-Spp
        - Paramecium-Spp
        - Uronema-Nigricans
      - Sessile
        - Sessile-Forming-Colonies
          - Contractile-Stalk
            - Carchesium
            - Zoothammium
          - Rigid-Stalk
            - Epistylis-Spp
            - Opercularia-Asymmetrica
            - Opercularia-Spp
        - Sessile-Individual
          - Vorticella-Convallaria
          - Vorticella-Infosionum
          - Vorticella-Microstoma
          - Vorticella-Similis
          - Vorticella-Spp
    - Carnivorous
      - Carnivorous-Free-Swimming
        - Coleps-Hirtus
        - Litonotus-Spp
      - Carnivorous-Sessile
        - Acineta-Spp
        - Discophyra-Spp
        - Podophyra-Spp
        - Tokophyra-Spp
    - Unidentified-Ciliates
  - Flagellates

```
                  Flagellates>20micron
                  Zoo-Flagellates<20micron
         Total-Filaments
            Filam-T-At
   Descriptor-Quantitative
      Ammonia
      Biochemical-Oxygen-Demand
      Chemical-Oxygen-Demand
         Cod-E
         Cod-I
         Cod-P
      Chloride
      Conductivity
      Greases
      Inhibitors
      Metals
      Mixed-Liquor-Suspended-Solids
         MLSS-At
         MLSS-Ras
      Volatile-Suspended-Solids
         Mixed-Liquor-Volatile-Suspended-Solids
      N-Total
         Tn-E
         Tn-I
      Nitrate
      Nitrite
      Oils
      Phosphate
      Phosphorous
      Temperature-T
      Total-Kjeldahl-Nitrogen
      Total-Suspended-Solids
         TSS-E
         TSS-I
         TSS-P
      Turbidity
      V30
On-Line
   Dissolved-Oxygen
      Do-At
   Ph
   Red-Ox
   Sensor-Temperature
   Sludge-Flow-Rate
   Water-Flow-Rate
```

```
                    Water-Flow-Rate-By-Pass
                    Water-Flow-Rate-E
                    Water-Flow-Rate-I
                    Water-Flow-Rate-P
  Water-Feature
```

This hierarchy, whose top-level categories are snapshot in Figure 6.1 can be used for several purposes (Sowa 2000):

- for reasoning: the categories support inheritance of properties from supertypes to subtypes;
- for queries: the categories map to the object of the database;
- for language analysis: the categories determine the constraints on permissible terminology.

Yet this hierarchy is specialized for a single application: *Storm* is considered an *Operational-Problem*, *Eukaryotic-Biological-Living-Object* and *Bacterium* are subtypes of *Wastewater-Biological-Living-Object*. For WaWO, the restrictions illustrated in its hierarchy simplify the inference engine that computes the answers of OntoWEDSS. But the simplifying assumptions that are convenient for OntoWEDSS would cover or eliminate details that might be essential for other applications. Reducing *Metazoa* to only *Nematode* and *Rotifer* makes, for instance, the ontology unusable for general biological analysis. Moreover, different applications may classify the same objects in very different ways and an ontology that is optimized for one application may make knowledge sharing and reuse difficult or impossible. Nevertheless, as noted above, the WaWO ontology is created to be easily reusable, at least in applications used in the wastewater domain.

6.3 Ontolingua vs. OIL: strategies for achieving a standard

In general there are two strategies for achieving a standard for an interchange language: defining a large set of modelling primitives that are present in some of the approaches of a community and glue them together; or defining a small set of modelling primitives that are common across the community, and defining a proper semantics for them. Both may lead to success. The first approach has been taken, for example, by Ontolingua and the UML[1] community. UML is broad enough to cover all its modelling concepts. This leads to ambiguity and redundancy in modelling primitives and sometimes a precise semantic definition is lacking. However, UML has been adopted by the software industry as one of the major approaches in the meantime and is therefore also a success. The second approach can be illustrated with OIL and HTML. The first version of HTML was very simple and limited but allowed the Internet to catch on and became a worldwide standard. Now we have HTML version 5, XHTML[2] and XML. So, beginning with a core set and successively refining and extending it has proven to be a successful strategy, too. Therefore, these two opposite approaches to standardization may both work successfully (Fensel *et al.*).

Ontolingua[3] (Gruber 1993; Farquhar *et al.* 1997) is an existing proposal for an ontology interchange language. It was conceived to support the design and specification of ontologies with a clear logical semantics based on KIF (see §2.4.6). Ontolingua extends KIF with additional syntax and a Frame Ontology to define object-oriented and frame-language terms. The Ontolingua Server as described in

[1]The Unified modelling Language (UML) is a language for specifying, visualizing, constructing and documenting the artifacts of software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

[2]XHTML 1.0 is the eXtensible HyperText Markup Language, a reformulation of HTML 4 in XML 1.0.

[3]http://ontolingua.stanford.edu/ and §2.4.7

Farquhar *et al.* (1997) has extended the original language by providing explicit support for building ontological modules that can be assembled, extended, and refined in a new ontology. The set of KIF expressions that Ontolingua allows is defined in the Frame Ontology. The Frame Ontology specifies, in a declarative form, the representation primitives that are often supported with special-purpose syntax and code in object-centered representation systems (e.g., classes, instances, slot constraints). Ontolingua definitions are LISP-like forms that associate a symbol with an argument list, a documentation string, and a set of KIF sentences labelled by keywords. An Ontolingua ontology is made up of definitions of classes, relations, functions, objects, and axioms that relate these terms. Moreover, Ontolingua includes a reification mechanism, which allows the treatment of statements of the language as objects, thereby making possible to express statements over these statements. The problem with Ontolingua is its high expressive power, which is provided without means to control it. A related problem is that not much reasoning support is provided with Ontolingua. In fact, it may be possible to provide reasoning support for Ontolingua using ATP [4], but neither the system nor any proof of its correctness are available. Nevertheless, the Ontolingua Server has been and still is a highly successful. The Web availability allowed to reach a wide audience with a high quality, robust, useful and public tool. Furthermore, the software is supported, maintained and documented. Users can directly use and evaluate the software, without buying, downloading and configuring it. Users who are familiar with a Web browser can use the Ontolingua Server right away. Context-sensitive hypertext documentation is accessible through the Web interface. Finally, the Server is working smoothly most part of the time and access problems are minimal.

[4] ATP is a LISP implementation of a model elimination theorem prover for full first order logic as well as context logic. It is a small, but efficient theorem prover based on Stickel's PTTP (Prolog Technology Theorem Prover) method. It should be easy to extend and easy to embed in a larger application (http://www.ksl.Stanford.EDU/software/ATP/).

OIL[5] (Fensel *et al.*) takes the opposite approach with respect to Ontolingua. It has a very simple and limited core language, based on the strategy of reduced initial complexity and controlled (possibly important) extension when required. In OIL, the focus on different reasoning tasks may lead to different extensions (features), still with clearly defined semantics. The current expressiveness of OIL is not sufficient for some purposes (Klein *et al.* 2000) and this may lead to a family of controlled extensions to the language. This will generate versions with different expressive power which can be applied in different cases as required. Modelers will be free to use these language extensions, but it is clear that this may prejudice reasoning support. This approach is believed preferable to the definition of one single, large and more difficultly manageable language. The OIL approach stems from the purpose for which OIL is designed: it should provide machine understandable semantics of domain theories in the Internet context. Therefore, clear definitions of semantics and reasoning support are essential. But despite a promising specification, the OIL proposal is still young and lacks several important features. OIL provides a mechanism for inheriting values from super-classes, but such values cannot be overwritten; if an attempt is made at *overwriting* an inherited attribute value, this will simply result in inconsistent class definitions. In OIL there is no facility for describing arbitrary *axioms*; the lack of an axiom-language prevents *composite* definitions of relations. *Modules* management in OIL is identical to the Namespaces mechanism in XML; it amounts to a textual inclusion of the imported module, where name clashes are avoided by prefixing every imported symbol with a unique indicator of its original location; however, much more elaborate mechanisms would be required for the structured representation of large ontologies. OIL currently does not allow the *use of instances in slot-values* or the *extensional definition of classes* (i.e., class definition by enumerating the instances of the class), but this should not be

[5]See §2.4.7.

a serious restriction, as ontologies are, in general, independent of specific instantiations. OIL currently does not support *concrete domains* (such as integers, reals and strings) and extensions in this direction are probably required; these extensions can compromise the decidability of the language and corresponding extensions to OIL's reasoner would be required if reasoning support is to be provided. Finally, OIL has a very *limited second-order expressiveness* and recursion potential[6]; even if a full second-order extension would be undesirable (even unification is undecidable in second-order logic), weaker second-order constructions seem to be required and a precise characterization of such expressiveness is called for in a future extension.

6.4 WaWO's important features

The WaWO ontology is a model that defines the meaning of each term used in the wastewater domain, in a precise and as unambiguous manner as possible. WaWO links classes and individuals at least with taxonomic/hierarchical relations and provides a terminology that each agent involved can understand and use. The *basic categories* in the WaWO model are represented as *objects* with specific *properties* and *relations*. Objects are structured into a taxonomy and the definitions of objects, slots and relations are specified according to the Ontolingua version of the Frame Ontology. Objects include classes referring to the general wastewater domain, to decision support, as well as to a detailed micro-organisms taxonomy. In this way, WaWO integrates the knowledge about WWTP management with the current knowledge about microbiological processes. A complete list of WaWO's features is presented in the Appendix §A.

Frame-based formalism. Being WaWO stored at the Ontolingua KSL Server (see §2.4.7), we implicitly chose a *frame-based formalism* for the Ontology, and this constraints the knowledge representation. While the central modelling primitives of

[6]Only classes are provided, not meta-classes nor individuals.

predicate logic are predicates, frame-based and object-oriented approaches take a different point of view. Their central modelling primitives are classes (i.e., frames) with certain properties called attributes. These attributes do not have a global scope but are only applicable to the classes they are defined for (they are typed) and the *same* attribute name may be associated with different value restrictions when defined for different classes. (Inherited attribute values can be overwritten.) A frame provides, therefore, a certain context for modelling one aspect of a domain[7]. The Ontolingua formalism paradigm used plays an important role in ontology integration, too[8]. WaWO integrates within the OntoWEDSS system, and specifically with a RBES and a CBRS. To be embedded into the OntoWEDSS system, WaWO will be manually translated to Allegro Common LISP.

Axioms. WaWO implements the semantics in a set of *axioms* (or rules)and relations that also enable to describe wastewater processes, to automatically deduce the answer to several questions about the wastewater domain, such as cause-effect questions, and to solve diagnosis impasses. Through the definition of axioms we are able to represent two kinds of *cause-effect relations*. We can associate micro-organisms to the problematic situations that they cause, and we can associate the actual state of the plant to the actions that need to be performed in order to reach the normal state from that actual state. Axiom deductions should be determined by a set of questions used to decide the competence of WaWO representation. Since there does not exist a standard for determining the competence of a model, we will define a set of questions about wastewater processes and the axioms used to answer them. In WaWO, axioms connect the descriptors to the states of the WWTP. More general states are linked to the descriptors by more general axioms, while less general states

[7]Many frame-based systems and languages have been developed, and under the name *object-orientation* the paradigm has also conquered the software engineering community.

[8]For example, to integrate an ontology built using a language based on first-order logic into WaWO, a lot of knowledge could be lost due to the weaker expressive power of the frame-based paradigm.

or sub-states are linked to the descriptors by more specific axioms. In the case of micro-organism descriptors, this means linking the state of the plant to higher or lower levels of micro-organism taxonomy. (Example: in case of presence of two micro-organism *species* and lack of axioms (or discordant axioms), the connection to the state of the WWTP is done at the taxonomic-*class* level.) In this way, we endow the ontology with *inference processes* and we improve the diagnosis and the decision support (see also §5.4.2 and §5.4.3).

Uncertainty. A realistic representation of physical facts should always contain some variable representing uncertainty, a margin of error which is due to the limited precision of our instruments. A zero uncertainty expedites knowledge management in many cases, but may cause logical inconsistencies in the ontology and contradictions in inferences. Another theoretical problem due to the representation of a physical system with a zero uncertainty (infinite precision), in addition to the logical inconsistencies that might arise, is the inability to represent infinite precision in a computer. In the general case, this may be even more troublesome than the logical inconsistency and this is another reason to try to find a physically realistic representation of physical quantities, including time. In the case of WaWO, however, this is a secondary problem, mainly because the ontology does not represent time. In WaWO, we just relax the degree of uncertainty associated to the diagnostics of the RBES and the CBRS, as will be explained in §8.5.

Time representation. We think that a representation containing time points and closed intervals on the real line is the easiest to understand, and for that reason we would prefer such a system for WaWO. Even if this feature is not implemented, we believe that WaWO's initial time-representation should be made as simple as possible to learn and use, even if alternative representations have some greater conceptual elegance. One question of interest is whether we can find some set of axioms that will allow (maintaining the consistency of inferences) more than one of the basic time theories to be used for descriptions of environmental processes and events, where an uncertainty variable always greater than 0 can be included as an element of every assertion about physical quantities, including time.

Categories counting. In October 2001, WaWO class hierarchy has 300 *classes* defined. For a comparison (see Table 6.1), the average number of definitions (classes, relations, functions, axioms and individuals) of the ontologies loaded in the Ontolingua Server is less than 100 and the UpperCyc ontology has 1618 classes.

6.5 Methodology overview

The WaWO ontology construction followed these steps:

1. Study of the *potential utility and the purposes* of the ontology (see §6.1).

2. *Knowledge acquisition process* (see Comas *et al.* (1999)): we were able to reuse some existing specific encoded knowledge and for the rest we went through interviews with experts and texts analysis.

3. *Conceptualization step*: characterization of the domain knowledge through a set of standard features (Arpírez Vega *et al.* 2000), identification of a preliminary list of the objects in our domain of discourse (the wastewater) (see §A.2 for a list of higher-level concepts), and identification of the properties of these objects and the relations that exist over them.

4. *Ontological-engineering environment*: we chose as development environment for the ontology the quasi standard Ontolingua framework.

5. Decision about which other existent ontologies to *reuse* in the creation of WaWO.

Table 6.1: Part of the Ontolingua library

Abstract-Algebra	18 December 2000	(25 definitions)
Agents	22 October 2000	(8 definitions)
Basic-Matrix-Algebra	18 December 2000	(26 definitions)
Bibliographic-Data	18 December 2000	(79 definitions)
Chemical-Crystals	18 December 2000	(121 definitions)
Chemical-Elements	18 December 2000	(170 definitions)
Cml	18 December 2000	(96 definitions)
Component-Assemblies	18 January 2001	(15 definitions)
DAML-Ont-Ontology	12 January 2001	(50 definitions)
Device-Ontology	30 September 1997	(28 definitions)
Documents	28 March 2000	(31 definitions)
Enterprise-Ontology	18 December 2000	(177 definitions)
Frame-Ontology	11 January 2001	(67 definitions)
Hp-Product-Ontology	30 September 1997	(124 definitions)
HPKB-Upper-Level-Kernel-Latest	18 January 2001	(2180 definitions)
HPKB-Upper-Level-Latest	18 January 2001	(2819 definitions)
HPKB-Upper-Level-Relations-Latest	18 January 2001	(639 definitions)
Interface-Definition-Language	18 January 2001	(27 definitions)
Interface-Ontology	18 January 2001	(51 definitions)
Job-Assignment-Task	18 January 2001	(20 definitions)
Kif-Lists	18 January 2001	(22 definitions)
Kif-Meta	18 January 2001	(39 definitions)
Kif-Numbers	18 January 2001	(89 definitions)
Kif-Relations	18 January 2001	(18 definitions)
Kif-Sets	21 February 2001	(31 definitions)
Mace-Domain	18 January 2001	(26 definitions)
Okbc-Ontology	8 March 2001	(47 definitions)
Parametric-Constraints	18 January 2001	(24 definitions)
Physical-Quantities	18 January 2001	(28 definitions)
Product-Ontology	28 January 1999	(46 definitions)
Simple-Time	18 January 2001	(254 definitions)
Slot-Constraint-Sugar	18 January 2001	(14 definitions)
Standard-Dimensions	18 January 2001	(45 definitions)
Standard-Units	18 January 2001	(61 definitions)
Tensor-Quantities	18 January 2001	(25 definitions)
Unary-Scalar-Functions	18 January 2001	(14 definitions)
Vt-Design	18 January 2001	(12 definitions)
Vt-Domain	18 January 2001	(769 definitions)
WaWO-Wastewater	12 September 2001	(350 definitions)

6. *Formalization in the Ontolingua language*: representation of objects by classes and individuals; representation of properties and relations by relations and functions in the Ontolingua language.

7. Definition of a set of *axioms* in to represent the constraints over the objects and properties in the ontology. This set of axioms constitutes a micro-theory (Lenat and Guha 1990) and provides a declarative specification for the various tasks we want to model.

8. Implementation of a *user interface* to access the ontology's content.

9. Assessment of the *scope and limitations* of the approach.

In this methodology it is possible to identify two meta-phases of the evolution of the ontology. One phase (steps 2-5 and 7) concerns the development of the conceptual structure of the ontology and the identification of: main concepts, taxonomies, relations, functions and axioms. Another phase (steps 6 and 8) involves formalizing the knowledge and introducing instances data into the conceptual structure. In the following sections we describe these two phases in detail.

6.6 Formalization of WaWO in the Ontolingua language

As seen in §6.4, after an exhaustive bibliographic research and taking into account the advises of other ontology engineers, we decided to adopt the Ontolingua formalism and development environment to build the WaWO ontology. Ontolingua is a complete work environment, which includes partial methodological guidelines, a specific language and an on-line editor (accessed through an HTML interface at http://WWW-KSL-SVC.stanford.edu).

The meta-ontology used by Ontolingua is the Frame Ontology which is used by default by all the ontologies being built on the server. The object of the wastewater domain are represented as classes and individuals. In more general terms, these classes and individuals, as well as functions, relations and axioms, are all frames (in the context of the meta-ontology).

The Ontolingua *development environment* has been described in §2.4.7. An excerpt of the resulting Ontolingua-package-LISP source code is the following (WaWO-wastewater.lisp):

```
(In-Package "ONTOLINGUA-USER")

(Define-Ontology
     WaWO-Wastewater
     (HPKB-Upper-Level-Kernel-Latest Frame-Ontology)
   "This is a specific ontology of wastewater treatment and
     disposal. The domain includes the microbiology of
     activated-sludge treatment plants. About the domain: The
     predominant method of wastewater disposal in large cities and
     towns is discharge into a body of surface water. Suburban and
     rural areas rely more on subsurface disposal. In either case,
     wastewater must be purified or treated to some degree, in
     order to protect both public health and water quality. This
     ontology will deal only with purification of wastewater
     discharged into a body of surface water.

     ...

     Author: Luigi Ceccaroni (UPC, Spain).

   ":Io-Package
   "ONTOLINGUA-USER"
   :Generality
   :Very-High
   :Maturity
   :Very-High
   :Shadow
   (Body-Of-Water City))

(In-Ontology (Quote WaWO-Wastewater))

;;; Compact

(Define-Individual Compact (Structure-Of-Floc) :Documentation
                   "Not supplied yet.")
```

```
;;; %-BOD-Removal

(Define-Class %-BOD-Removal (?X) :Def (Calculated-Descriptor ?X)
          %:Documentation "% removed after primary, secondary or
          %overall treatment.")

;;; Foam-Presence-Aeration-Tank

(Define-Function Foam-Presence-Aeration-Tank (?Frame) :-> ?Value
                 "Synonyms: ESC-B (cat)" :Def (And (Foam-Presence
                 ?Frame) (String ?Value)))

;;; Mixed-Liquor-Suspended-Solids

(Define-Okbc-Frame Mixed-Liquor-Suspended-Solids :Frame-Type
:Class
                   :Direct-Superclasses (Descriptor-Quantitative)
                   :Direct-Types (Set-Or-Collection Primitive)
                   :Own-Slots ((Documentation "Synonyms: Mixed liquor
                   suspended solids MLSS
                   Measure of biomass concentration") (Arity 1)))

;;; Storm

(Define-Okbc-Frame Storm :Frame-Type :Class :Direct-Superclasses
                   (Meteorological-Conditions-Adverse) :Direct-Types
                   (Set-Or-Collection Primitive) :Own-Slots
                   ((Documentation "Synonyms: Tempesta (cat) Tormenta
                   (cat)") (Arity 1)))

;;; Primary-Clarifier

(Define-Individual Primary-Clarifier (Sampling-Locations)
"Synonyms:
                   Sedimentation tank
                   Suspended solids that pass through screens and
                   grit chambers are removed from the sewage in
                   sedimentation tanks. These primary clarifiers
                   provide about two hours of detention time for
                   gravity settling to take place. As the sewage
                   flows through them slowly, the solids
                   gradually sink to the bottom. The settled
                   solids (known as raw or primary sludge) are
                   moved along the tank bottom by mechanical
```

```
                    scrapers. ... ")

;;; TSS-Inflow

(Define-Function TSS-Inflow (?Frame) :-> ?Value "Synonyms:
                 SST aigua entrada (cat)" :Def (And
                 (Total-Suspended-Solids ?Frame)
                 (Number ?Value)))

;;; Inflow

(Define-Individual Inflow (Sampling-Locations) "Not supplied
yet.")

;;; TSS-Exit-Primary

(Define-Function TSS-Exit-Primary (?Frame) :-> ?Value "Synonyms:
                 SST aigua entrada (cat)" :Def (And
                 (Total-Suspended-Solids ?Frame) (Number ?Value))
                 :Documentation "Synonyms:
                 SST sortida primary (cat)")

...

;;; Primary-Effluent

(Define-Individual Primary-Effluent (Sampling-Locations)
    "Synonym: Secondary-Inflow ")

;;; Recycle-Sludge

(Define-Individual Recycle-Sludge (Sampling-Locations)
                   "Not supplied yet.")
```

6.7 Reuse of existing ontologies

For the construction of large-scale, knowledge-based systems, it is essential that researchers are able to share and reuse representational components built by others. However, despite the potential advantages of such sharing, and the availability of such components in component libraries such as Ontolingua, it remains a challenging task to import and use such components.

There are still only a few published examples of such reuse (e.g., Borst (1997)[9], Borst *et al.* (1997), McGuire *et al.* (1993), Cutkosky *et al.* (1993)). There are also few published examples (e.g., Uschold *et al.* (1998)) describing detail of how ontologies are applied, in cases where an ontology is reused, (e.g. as the basis for building another ontology rather than starting from scratch).

In WaWO we (re)use the Frame Ontology, as it is the default meta-ontology in the Ontolingua environment. We also tried to reuse the UpperCyc ontology, the Ontolingua translation of the most general part of Cyc ontology. UpperCyc, at the moment, includes 1700 classes but does not cover most of the top concepts of WaWO, even if they are very basic (e.g., *Wastewater*, *Bacterium*, *Descriptor*). Reusing UpperCyc would not be a difficult task, since both WaWO and UpperCyc are written in Ontolingua Language and are stored in the library of ontologies at the Stanford KSL Ontology Server (Farquhar *et al.* 1995). Nevertheless, the sole fact that the *Wastewater* class is not included in UpperCyc make the connection between the two ontologies absolutely useless: the heterogeneity of the two lexicons is too high. In other words, the two ontologies, even if they share the same server, have a too different level of detail.

6.8 Reasoning and problem-solving

Most existing ontologies, including WaWO, are domain ontologies, reflecting the fact that they capture domain knowledge about the world independently of its use. However, one can (and sometimes have to) also view the world from a *reasoning* (i.e., *use*) perspective. For instance, if one is concerned with diagnosis, he will talk about

[9] Borst's Ph.D. thesis investigates on what happens when, instead of an ontology about the knowledge for simulation of technical devices, an ontology for a totally different task in a different engineering domain is constructed. This new task is the ecological impact assessment of product disassembly. It is demonstrated that parts of PhysSys Ontology (Borst *et al.* 1997) can be reused and extended with other (reusable) ontologies to form a new ontology formalizing a novel approach to product disassembly analysis. A KBS called ProMoD has been developed based on the new ontology. ProMoD serves as a prototype for a future extension of commercial software for ecological impact assessment.

hypotheses, *symptoms* and *observations*. We say that those terms belong to the *task ontology* of diagnosis. Similarly, one can view the world from a problem-solving point of view, and see the world in terms of *states*, *state transitions*and *preferences*. These terms are part of the *method ontology* of the problem-solving domain.

When working with an ontology, several reasoning means are available. Subsumption checking is one of the most important. Other possible reasoning tasks are, for example, instance classification, query subsumption and query answering over classes and instances, navigation through ontologies. However, many of them can be reformulated in terms of subsumption checking.

Reasoning in ontologies is defined by axioms. As shown in §5.4, in WaWO, axioms about cause-effect relations between microbe presence and state of the plant and between state of the plant and actuation can generate a certain degree of dynamism. Inferences are started under petition of other elements of OntoWEDSS; if inferences are successful, these petitions are positively answered, and the answers explained.

But in wastewater domain the relations among concepts are not completely well-known. Sometimes the experts do not agree on the structure of the domain. Although various mathematical models have been introduced to describe the relation between micro-organisms and substrate and between micro-organisms and states of the WWTP, they do not provide an entirely satisfactory description of the cause-effect links existing within the treatment plant. For this reason, the inferences of the ontology on cause-effect axioms are limited by the actual knowledge available to the experts at the moment of axiom encoding.

6.9 Knowledge sharing

The potential sharing of acquired knowledge bases, the last research objective of this thesis is achieved by the introduction of a formal ontology, such as WaWO. Two central problems exist in achieving a real sharing: heterogeneous representation languages and heterogeneous ontologies.

Heterogeneous representation languages. There is no single knowledge representation that is best for all problems, but there are growing standards, often helped in their diffusion by growing Internet access. Another issue is that the choice of one form of knowledge representation over another can have a big impact on a system's performance. Thus, in many cases, sharing knowledge will involve translating from one representation to another. More tools are needed that can help automate the translation process.

Heterogeneous ontologies. Even if the representation problems are resolved, it can still be difficult to combine two knowledge bases or establish effective communications between them. The absence of a shared vocabulary presents a further barrier, which could be removed through the development of shared sets of explicitly defined terminology (e.g., ontologies). For ontologies to be most useful, the definitions provided should include declarative constraints that specify the semantics of the terms being defined, and procedural methods, that enforce those constraints when the terms are used in an application, should be provided.

With WaWO we want to overcome these impediments to knowledge sharing in the wastewater domain, both from a theoretical and a practical point of view, even if the complete solution of these problems is necessarily outside of the scope of this thesis.

A first effort in the development of practical applications for knowledge sharing, apart from the implementation in Allegro Common LISP, has been using the WaWO knowledge base the to build a customized glossary, to be employed in Babylon Tool (version 3.1), a growing standard in on-line and off-line translators, dictionaries and converters. The glossary we built is called *Wastewater (WaWO)* and was submitted to the Babylon Index and published on the Babylon site. To access the glossary, it is sufficient to go to the Babylon page[10], search for *wastewater* at the *glossary index* window and then subscribe it, for on-line use, or download it for off-line use. The glossary *Wastewater (WaWO)* can be also found under the following categories: 1. Ecology, 2. Biology.

[10] http://babylon.com

Chapter 7

Implementation

A pesar de lo que digan, la idea de un cielo habitado por Caballos y presididos por un Dios con figura equina repugna al buen gusto y a la lógica más elemental, razonaba los otros días el Caballo.
Todo el mundo sabe -continuaba en su razonamiento- que si los Caballos fuéramos capaces de imaginar a Dios lo imaginaríamos en forma de Jinete.

Augusto Monterroso

In this chapter, the implementation of OntoWEDSS is described in detail. OntoWEDSS has been designed as a modular and multi-layered architecture (see Figure 5.4) with 3 layers: perception, diagnosis and decision support. The implementation work of the thesis focuses on the diagnosis layer, whose three main modules are: the CBRS, the RBES and WaWO. However, the whole architecture of OntoWEDSS will be described, starting from the perception layer (§7.1) which includes data analysis, interpretation and management. Then the focus will be on the description of the implementation of the RBES module, the CBRS module and the ontology, i.e. the diagnosis layer (§7.2). Finally, the third layer, involving decision support, the reasoning integration (§7.4) and the supervisory working cycle are described.

7.1 Perception layer

The perception layer is where the process of information integration begins. From the information modelling point of view, two integration levels can be distinguished and have to be dealt with in order to achieve a completely integrated access to information (Stuckenschmidt 2000). The first level is the *structural integration*, which is concerned with network technology and communication protocols, ensuring that the different information sources can physically communicate. In the last decade, the *Internet* has established a stable infrastructure for exchanging large amounts of information. A widely shared and stable set of protocols (e.g., TCP/IP, HTTP, FTP) now make it possible to access information from Web-connected data-bases, and Web-enabled programs. Once sources can physically exchange information, they must agree on a common syntax for exchanging such information (*syntactic integration*). Later in this section, we will deal with another kind of combining information: the temporal integration.

An example of the list-based syntaxis of input data in OntoWEDSS is:

```
("Wastewater treatment plant of Granollers" 21
  ("Water-Flow-Rate-I" "quantitative" 0.6 3
    "0" "19000" "Low"
    "19001" "24000" "Normal"
    "24001" "60000" "High")
  ("TSS-I" "quantitative" 0.7 3
    "0" "230" "Low"
    "231" "330" "Normal"
    "331" "750" "High")
  ("COD-I" "quantitative" 0.8 3
    "0" "650" "Low"
    "651" "850" "Normal"
    "851" "2850" "High")
  ("TN-I" "quantitative" 0.9 3
    "0" "60" "Low"
    "61" "90" "Normal"
    "91" "200" "High")
  ("TSS-P" "quantitative" 0.7 3
    "0" "40" "Low"
```

```
  "41" "120" "Normal"
  "121" "500" "High")
("COD-P" "quantitative" 0.9 3
  "0" "530" "Low"
  "531" "720" "Normal"
  "721" "1500" "High")
("MLSS-AT" "quantitative" 0.7 3
  "0" "1000" "Low"
  "1001" "2000" "Normal"
  "2001" "4000" "High")
("SVI-AT" "quantitative" 0.8 3
  "0" "50" "Low"
  "51" "100" "Normal"
  "101" "1000" "High")
("SRT-AT" "quantitative" 0.7 3
  "0" "5" "Low"
  "5.1" "8" "Normal"
  "8.1" "15" "High")
("F-M-AT" "quantitative" 0.7 3
  "0" "0.25" "Low"
  "0.26" "0.35" "Normal"
  "0.36" "1.5" "High")
("Filam-Dominant-AT" "qualitative" 1 0.8 20
  "Haliscomenobacter-Hydrosis"
  "Microthrix-Parvicella"
  "Gordona"
  "Nostocoida-Limicola-I"
  "Nostocoida-Limicola-II"
  "Nostocoida-Limicola-III"
  "Thiothrix"
  "Type-0041"
  "Type-0092"
  "Type-021-N"
  "Type-0411"
  "Type-0581"
  "Type-0675"
  "Type-0803"
  "Type-0914"
  "Type-0961"
  "Type-1701"
  "Type-1863"
  "Zoogloea-Ramigera"
  "?")
("DO-AT-1" "quantitative" 0.5 3
  "0.0" "2.0" "Low"
```

```
  "2.1" "4.0" "Normal"
  "4.1" "100" "High")
("DO-AT-2" "quantitative" 0.5 3
  "0.0" "2.0" "Low"
  "2.1" "4.0" "Normal"
  "4.1" "100" "High")
("TSS-E" "quantitative" 0.8 3
  "0" "10" "Low"
  "11" "20" "Normal"
  "21" "95" "High")
("COD-E" "quantitative" 0.8 3
  "0" "110" "Low"
  "111" "140" "Normal"
  "141" "465" "High")
("TN-E" "quantitative" 0.9 3
  "0" "60" "Low"
  "61" "90" "Normal"
  "91" "200" "High")
("Diagnosis" "qualitative" 1 1 35
      "Deflocculation"
      "Electrical-Blackout"
      "Foaming-Sludge"
      "Foaming-Sludge-Actinomycetes"
      "Foaming-Sludge-Microthrix"
      "Foaming-Sludge-Gordona"
      "Hydraulic-Overloading"
      "Incomplete-Nitrification"
      "Mechanical-Problem"
      "Normal-Operation"
      "Organic-Overloading"
      "High-Solids-Loading"
      "Hydraulic-Shock-Primary-Treatment"
      "Inadequate-Sludge-Purge"
      "Low-Efficiency-Of-Grit-Removal"
      "Old-Sludge"
      "Primary-Clarifier-Problem"
      "Pumps-Or-Pipes-Blocked"
      "Septic-Sludge"
      "Sludge-Removal-Systems-Break"
      "Too-High-Sludge-Density"
      "Bulking-Sludge-Filamentous"
      "Bulking-Sludge-Slime-Viscous"
      "Dispersed-Growth"
      "Pin-Point-Floc"
      "Rising-Sludge"
```

```
        "Toxic-Shock"
        "Aeration-Problem"
        "Hydraulic-Shock-Secondary-Treatment"
        "Imbalanced-Flow-Rate"
        "Electrical-Problem"
        "Storm"
        "Secondary-Clarifier-Problem"
        "Surfactant-Scum"
        "Underloading"
        "?")
  ("WAS" "quantitative" 0 1
    "0" "20000" "control-descriptor")
  ("RAS" "quantitative" 0 1
    "0" "300" "control-descriptor")
  ("Air-Flow-1" "quantitative" 0 1
    "0" "300000" "control-descriptor")
  ("Air-Flow-2" "quantitative" 0 1
    "0" "200000" "control-descriptor"))
```

The domain of OntoWEDSS is represented by a set of descriptors about wastewater treatment. The perception layer of OntoWEDSS collects all data about the WWTP, in order to update all descriptors' values in a living database. Data are distinguished in three types: on-line data coming from sensors and other equipment, off-line data (including quantitative and qualitative descriptors) and calculated data.

1. *On-line data* are simulated because OntoWEDSS is not physically connected to a WWTP. On-line data are always quantitative. When connected, on-line data will include hundreds of digital signals about the mechanical equipment of the specific WWTP (e.g., pump engines switched on and off, blowers functioning at high and low velocity, floodgates open and closed). Apart from these specific signals, on-line data will also include about 20 general analogical signals generated by sensors: flow rates (at inflow, primary effluent, effluent, by-pass, RAS, WAS, aeration-tank effluent and AT recirculation *sampling-points*) and physical descriptors, such as pH and dissolved oxygen concentration at each compartment of the aeration tank. On-line data acquisition will be directly

accomplished by the PLCs network of the WWTP. These PLCs will be connected to a master PLC, which transmits, every 15 seconds, the whole data batch to the computer where OntoWEDSS resides. A specific program will be in charge of communication with the master PLC.

2. *Off-line data* can be quantitative and qualitative. Quantitative ones are provided by the results of analytical determinations on samples collected daily from different locations of the WWTP. Qualitative data come from daily *in situ* observations, which are recorded in the laboratory database once a day and are composed of the following descriptors: plant performance, quality of biomass, settling characteristics. Qualitative descriptors can be ordered or non ordered. Off-line data, in general, are read by OntoWEDSS as an ASCII file. There are, though, several microbiological descriptors, whose values are determined only once a week. These data are introduced into OntoWEDSS by the user through the API of the RBES module. Figure 7.1 on the next page shows the dialog box to manually introduce descriptors' values into the OntoWEDSS system.

   A module to support the identification of filamentous organisms and microfauna by uneducated operators can be integrated to the perception layer, following the example illustrated in Comas (2000). It will ask for evident (observable with a microscope) organism-descriptors, such as the presence of branches in the filaments, the cell's shape, the presence of *flagella*, *cilia* or *pseudopodia* in protozoan, the existence of colonies. This module can be interactive and, once a micro-organism is identified, it can give the user extra-information about the significance of its presence.

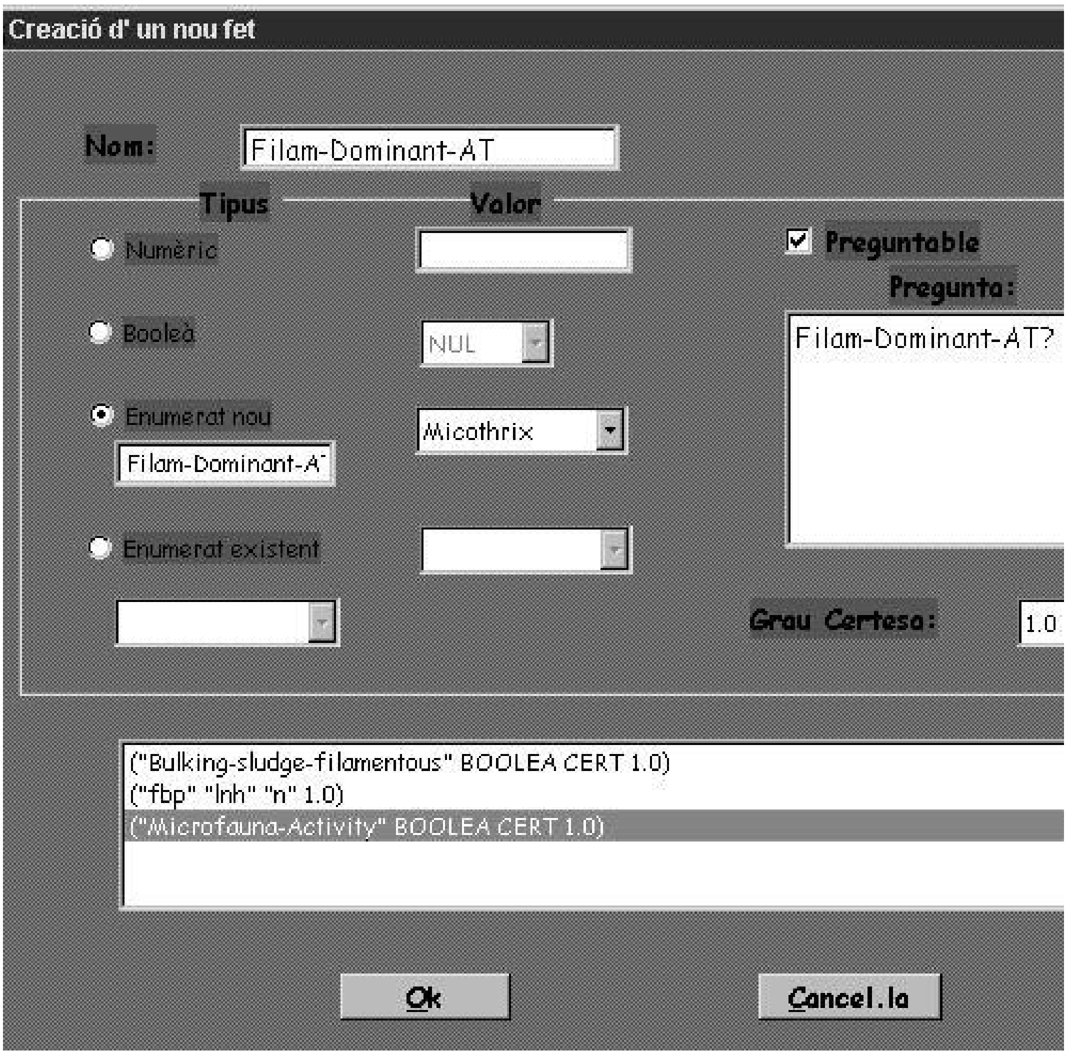

Figure 7.1: Window for the introduction of microbiological data through the API of the RBES of OntoWEDSS.

3. *Calculated data* are combination of quantitative data, which allow the assessment of global-process descriptors, such as Sludge-Residence-Time, Sludge-Volumetric-Index, Food-To-Micro-Organism-Ratio, percentage of COD, BOD and TSS removal by primary, secondary and overall treatments. With respect to micro-organisms information, the relative abundance of microbiological species is calculated (assigning to each species-descriptor one of the following ordered values: none, few, some, equilibrium, abundant, excessive), as well as are the predominant filamentous organism and the predominant protozoan.

Raw data require a number of processing procedures to be validated, integrated into a uniform time-scale, discretized and recorded into the database, before being used. Each one of these procedures will now be described.

Data validation includes checking for existence, for range-correctness of values, for validity-expiration and gathering periodicity. There is a procedure that detects whether data exist in a machine-readable form (this usually depends on whether data are measured or not). A procedure checks if the values of the data are within permitted ranges. Out-of-range values are deleted (e.g., negative values for chemical analyses of wastewater and sludge quality, or values for the *Water-Flow-Rate-I* that are higher than the maximum possible flow-rate for a specific WWTP). A procedure updates the data values, when their validity time expires. Once this time has passed, a new value has to be acquired by the system and stored into the database. On-line data, for example, are updated every 15 seconds, while off-line data have different validity times (e.g., 36 h for Inflow-COD and 3 d for Filamentous-Dominant).

Data temporal-integration accommodates different rates of sampling into two uniform temporal units of 1 h and 1 d. Average values of data coming from sensors (water and sludge flow rates, DO, air flow and pH) are calculated. For all other data, the last value stored in the database is considered.

Data discretization is defined by a set of cuts over descriptors' domains. Quantitative descriptors are in general discretized using two cuts.

Data recording produces a hourly and a daily record of the plant state, in the form of two living databases, from which the reasoning modules (RBES, CBRS and WaWO) obtain the data. The two records include therefore data coming from sensors, analytical data, *in situ* observations, microbiological information and calculated data.

7.2 Diagnosis layer

The diagnosis layer is implemented in Allegro Common Lisp 5.0.1, a programming environment developed by Franz, Inc. This tool let generate an application in Common LISP with a graphical interface. The layer is composed of three modules: *rule-based expert system* (RBES), *case-based reasoning system* (CBRS) and *ontology* (WaWO).

7.2.1 RBES

This module is a shell enabling the development of an expert system based on rules (see §4.1). The user defines the set of data/facts (data base) and the set of rules (knowledge base). The diagnostic comes from the execution of the RBES with the data and rules introduced. An interaction with the user during execution is possible. The result is presented together with an explanation which shows the reasoning sequence. The elements which compose this module are the same of a typical RBES (see Figure 4.1): data base, KB formed by rules and meta-rules, inference engine, user interface with questions-answers management, explanation module which trace the reasoning of the system, and knowledge engineer interface which connects the engineer with the KB and the data base. In this section we first characterize each element of this module and then describe the specific implementation of uncertainty in the RBES (see also §5.4.1).

Data base. Facts define the current state of the system. They are dynamic and change due to rules application. Manually introduced and automatically read data are all kept in the data base, while inferred data are held in the working memory during execution. Facts include the following information:

- *Name.* It is a unique identifier in a data base.
- *Type.* It can be pre-defined, e.g. numerical and boolean, or user-defined. Users can define enumerated/qualitative types, specifying all possible values for a new type. Enumerated types can be classified as ordered and non ordered. For instance, the designer of the KB can define a new type named *LNH* with three values, specifying that they are ordered in the following way: *low*, *numerous*, *high*.
- *Interaction possibility.* It indicates if the value of the fact can be asked to the user when it is missing (*NIL*) and cannot be inferred from the knowledge in the expert system.
- *Question.* It is used by the system to directly ask the user about data values, if they are necessary to deduce new knowledge about the diagnosis process. The user interface allows this communication.
- *Value.* It can be *NIL* or anything coherent with the type. Data can take different values in time. The system asks the user for a value if the following conditions hold: (1) the value is missing, (2) it is necessary for deduction, (3) interaction is possible and (4) it cannot be deduced automatically from other facts and rules.
- *Degree of truth.* It is a numerical value within the [0, 1] interval.

Knowledge base. The KB characterize the heuristic knowledge of the domain. It is composed of three elements: rules, meta-rules and modules. The last two elements are optional.

- *Inference rules.* They have two main parts: an invocation condition which groups certain logical conditions (on the left side of the GUI window; see Figure 7.2) and an action (on the right side of the GUI window). The action is called *conclusion* if it specify a new assertion to be placed in the working memory. Otherwise, actions specify operations in the real WWTP, e.g. the opening of a valve. More extensively, rules are composed by the following elements:
  - *Name.* Unique identifier of a rule in a KB.
  - *Invocation condition* or *premise.* It is made of a set of conditions connected by the conjunction logical-operator:

    $\bigwedge_{i=1}^{n} condition_i \rightarrow action$

    Each condition has the following configuration:

    <Negation-state> `<fact-x>` <operator> `<value/fact-y>`

    where *negation-state* is an operator determining if the condition is or is not negated; *fact-x* corresponds to any fact of the data base; *operator* is a relation between *fact-x* and *value/fact-y* (e.g., $<$ or $>$); *value/fact-y* is a constant value or some fact of the same type of *fact-x*.
  - *Action.* It can be of two categories: conclusion and operation. An example of each category follows:
    - Conclusion. If *Filamentous-bacteria-presence*-is-*numerous* and *TSS-E*-is-*high* then *Bulking-sludge-filamentous*-is-true.
    - Operation. If *Bulking-sludge-filamentous*-is-true then open-*Valve-10*.

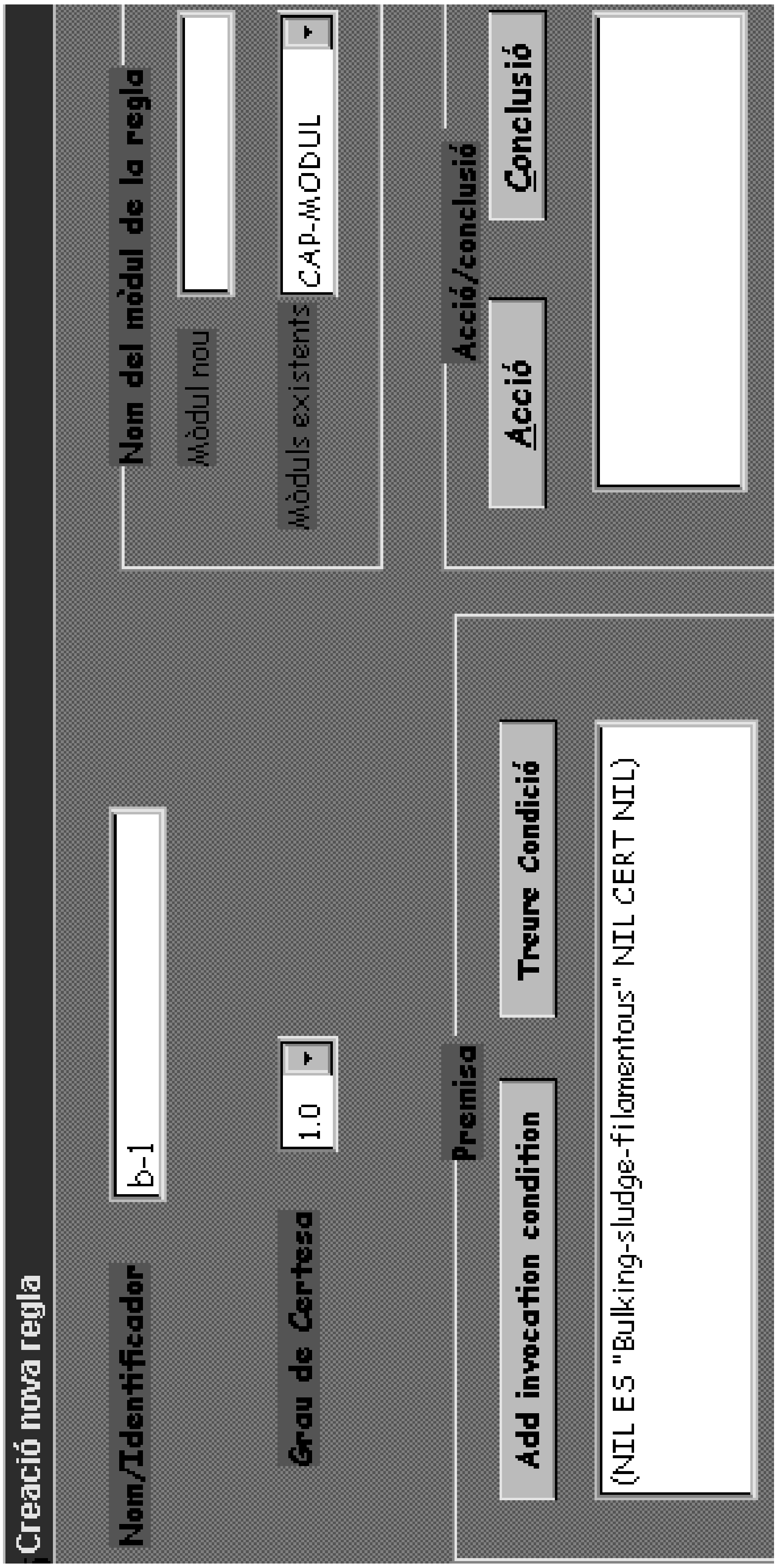

Figure 7.2: Window for the creation of rules in the RBES of OntoWEDSS.

  - *Certainty factor* of the rule. It is a numerical value in the range [0, 1].

  Example:

  - Name: *Rule-Filamentous-bacteria-presence*.
  - Invocation condition: *Filamentous-bacteria-presence* is *numerous* and *TSS-at-the-effluent* is *high*.
  - Action: *Bulking-sludge-filamentous* is true.
  - Certainty factor: 1.0.

- *Meta-rules*. They allow to guide the inference engine. The action of a meta-rule affects the inference process itself. According to the object of their action, *meta-rules* (MR) are classified in:

  - *Rules*-MR. They allow the activation and deactivation of a set of rules during the inference process.
  - *Modules*-MR. They determine if the type of search inside the modules is forward-chaining or backward-chaining.
  - *Strategies*-MR. They select the order in which modules are explored.
  - *Courses-of-action*-MR. They determine the order in which strategies are applied.

- *Modules*. They group a set of rules and meta-rules which have some common characteristic, such as similar actions or similar invocation conditions. The modules of a KB are usually organized in a hierarchical structure. The modularization improves the efficiency of validating and maintaining the KB. An example of how to modularize the KB of the RBES is to keep apart rules belonging to different diagnosis decision-trees, such as:

filamentous bulking	viscous bulking
deflocculation problems	rising-sludge
underloading problems	overloading problems
toxic shock	foaming
clarifier problems	primary treatment problems

Inference engine. It executes rules in a given order to solve a problem. Doing this, it can deduce new facts. An interaction with the user is possible through the user interface. The inference process can be of two different types (described in §4.1), but in both cases is divided in three steps:

1. *Detection.* The rules of the KB are evaluated matching the working memory against the invocation conditions. Applicable rules are chosen accordingly and stored as a new set.

2. *Selection.* From the set of applicable rules, one is selected for execution. The selection strategy can vary; here are a few examples: first rule in order, rule with the easiest evaluation, most/least used rule, most-specific/most-general rule, most informative rule (the one which generate the highest number of unknown facts), rule with the highest degree of truth.

3. *Application.* The selected rule is executed and its associated action is activated. As noted above, an action can be a conclusion or an operation. A conclusion can generate new sub-goals and modification in the working memory. The degree of truth of the conclusion is automatically calculated.

User interface. It allows the user to communicate with the RBES in a friendly way, via the selection of options from menus and buttons. This module has three functionality:

- the introduction of the data of the problem in question (see Figure 7.1);

- consulting the RBES about the state of facts or rules;
- asking the user to confirm actions or about data values.

When manually introducing facts about the wastewater domain into the data base, the use of the vocabulary of the WaWO ontology is mandatory to avoid inconsistencies.

Explanation module. It composes information about the reasoning process: it traces the chain of rules fired, their modules and the strategies used in the inference process. The sequence of rules which leads to the conclusion is shown to the user in an easily understandable language.

Uncertainty. As noted above, a rule has the form:

$$\bigwedge_{i=1}^{n} condition_i \rightarrow action$$

The CF of the action is calculated as:

<rule's CF> * <invocation-condition's DT>

where CF is the certainty factor and DT the degree of truth. The DT of the invocation condition is:

$$\min(\mathrm{DT}_{condition-1},\ \mathrm{DT}_{condition-2},\ ...,\ \mathrm{DT}_{condition-n})$$

The DT of a condition depends on the type of its components (the facts), the operator ($>, \geq, <, \leq, =, is, equal$) and the very enunciation of the condition, which can be:

$$< Fact >< operator >< Value > \text{ or}$$

$$< Fact - x >< operator >< Fact - y >$$

The DT of a fact has as range the [0, 1] interval, while the DT of a value is always 1. The membership functions defined on the input facts are applied to their actual values, to determine the degree of truth for each rule's condition. This degree of truth is sometimes referred to as the condition's *alpha*. If the resulting rule's invocation condition has a nonzero degree of truth (i.e., if the rule applies at all), then the rule is said to *fire*. The DT of the invocation condition of each rule is then applied to find the certainty factor of the action part of each rule. The final result is a fuzzy subset to be assigned to each descriptor involved in the action part of the rule.

7.2.2 CBRS

This module is a shell for the definition of a case-based reasoning system, which, as described in §4.2, is structured in five steps, each of which involves some library or agent or process. We now analyze the data structures required to implement those steps and how these structures are manipulated.

Descriptors library. To represent domain knowledge, descriptor-value pairs are used. To represent a problem, a set of descriptors, with their associated values, is used. These descriptors are a subset of WaWO's vocabulary (see §6.2). Depending on the particular WWTP, a different subset can be chosen. The structure of a descriptor is the following (see Figure 7.3):

- *Name.* This is the identifier of the descriptor, for instance: *Water-Flow-Rate-I*, *Appearance-surface*, *Biomass-color*, *Bubble-presence*. This identifier is given by the user and it is mandatory to use a subclass of the *WWTP-descriptor* class of the WaWO ontology, to avoid inconsistency.

- *Weight.* It is a real number, which represents the importance of the descriptor in the description of the domain. The weight is used to calculate similarities between cases.

- *Type.* A descriptor can be quantitative, when its values are continuous, or qualitative, when its values are discrete. Qualitative types can be classified as ordered and non ordered; quantitative ones are defined as ordered. A few examples follow:

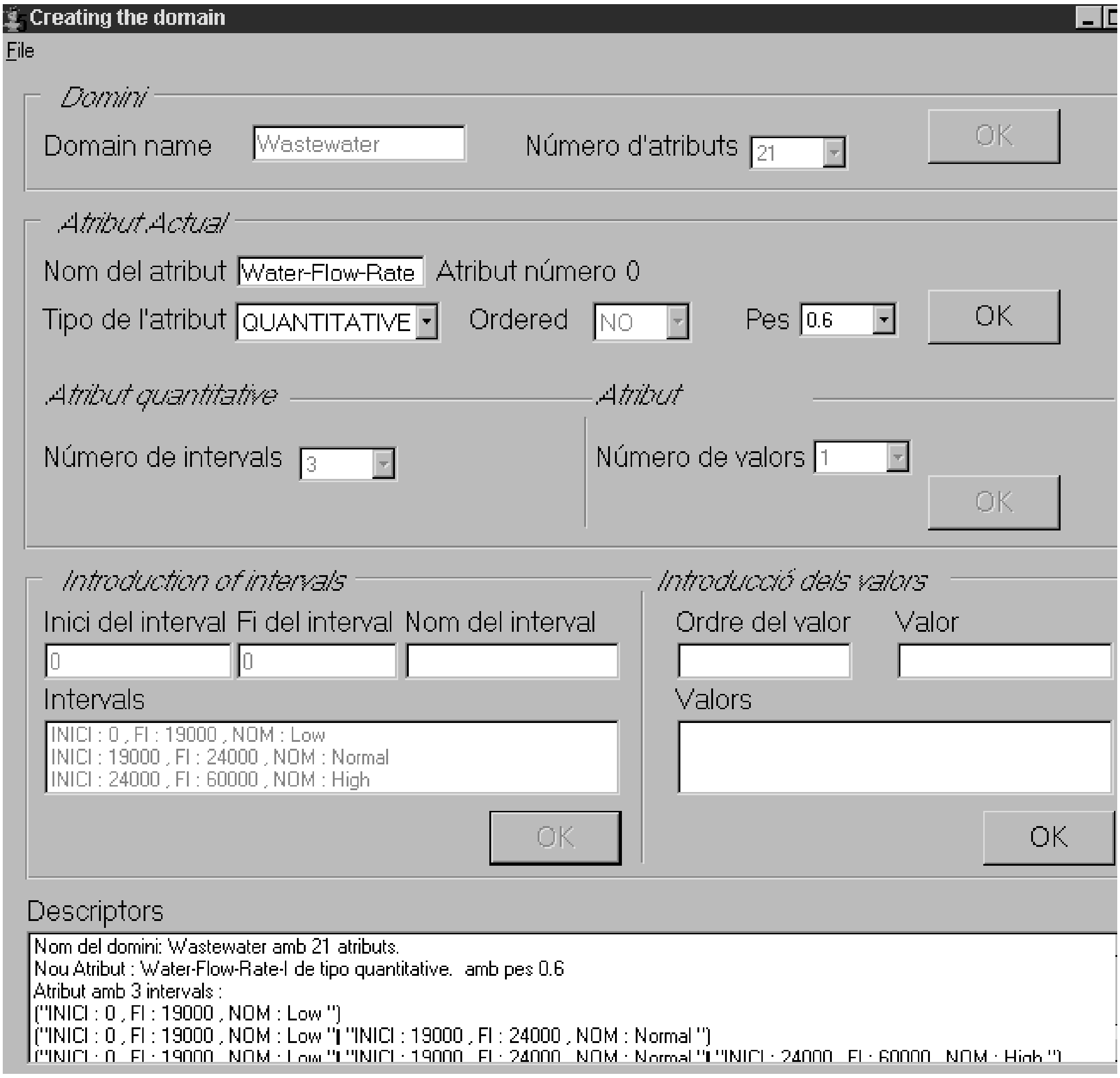

Figure 7.3: Window for the creation of a domain in the CBRS of OntoWEDSS.

- Quantitative: *Water-Flow-Rate-I* [0-60,000], *pH* [7-10], *BOD* [0-2,000], *COD* [0-3,000].
- Qualitative:
  - Ordered: *Structure-of-floc* {compact, slightly-dispersed, dispersed, very-dispersed}, *Gordona* {none, few, some, common, abundant, excessive}.
  - Non ordered: *Appearance-Floc* {small-floc-with-slow-settling, small-floc-with-fast-settling, large-flocs, poorly-defined-flocs, other-appearance}, *Settling-characteristics* {Good-settling-with-no-floating-sludge, fast-or-slow-settling-with-very-unclear-supernatant, fast-settling-with-moderately-unclear-supernatant, no-sedimentation, good-settling-with-floating-sludge, slow-settling-with-moderately-unclear-supernatant, dark-sludge, rising-sludge, good-settling-with-foams}, *Foam-presence* {white, brown, fatty, surfactant, none}.

- *Number of intervals or values.* Intervals are introduced to discretize continuous descriptors in a supervised way. The range of each interval is identified through a maximum value (real), a minimum value (real) and an identifying name (string); the range of qualitative descriptors is characterized specifying all possible values (strings).

- *List of intervals.* This list is defined only for quantitative descriptors, for instance:
  - Descriptor: *Water-Flow-Rate-I.*
  - Range: [0-60,000].
  - Intervals: 0-19,000: low; 19,000-24,000: normal; 24,000-60,000: high.

- *List of values.* This list is defined only for qualitative descriptors, for instance:
  - Descriptor: *Dominant-Filamentous-Bacteria.*
  - Ordered?: No.
  - Values: {Haliscomenobacter-Hydrosis, Microthrix-Parvicella, Gordona, Nostocoida-Limicola-I, Nostocoida-Limicola-II, Nostocoida-Limicola-III, Thiothrix, Type-0041, Type-0092, Type-021-N, Type-0411, Type-0581, Type-0675, Type-0803, Type-0914, Type-0961, Type-1701, Type-1863, Zoogloea-Ramigera}.

Action library. Actions are what the CBRS suggests as a reaction to a certain situation. Each action in the library has the following structure:

- *Action identifier.* It is a positive integer.
- *Action description.* It is exactly what OntoWEDSS will suggest to the user, in the case that CBRS's diagnostics is passed to the decision support layer; for instance:
  - "Check out *Food-To-Micro-Organism-Ratio*"
  - "Remove aeration-tank and clarifier foam"
  - "Reduce waste-activated-sludge flow rate (*FlowRate-WAS*)"
  - "Use minimum recycled-activated-sludge flow rate to facilitate good compacting"
- *List of descriptors on which the action depends.* An action is applicable only if all the descriptors on which it depends have a value.
- *Action's formula.* It is a function of the descriptors of the previous list.

- *Action's maximum value.* It specify a constraint on the action. For instance, if we are reducing the *FlowRate-WAS*, the constraint avoids en excessive reduction.

An example of how actions work is the following one:

- Action identifier: 1
- Action description: Set waste-activated-sludge flow rate (*WAS*)
- List of descriptors: *WAS*, *Water-Flow-Rate-I*
- Action's formula: $WAS_{new-case} = WAS_{retrieved-case} * \frac{Water-Flow-Rate-I_{new-case}}{Water-Flow-Rate-I_{retrieved-case}}$
- Action's maximum value: 20,000

In the *Case adaptation* paragraph below, we show a working example of use of an action's formula.

Case library. It represents the experience about the domain stored in OntoWEDSS. The case library stores only significant cases and not all of them. The structure of the case library can be hierarchical or sequential/flat. In OntoWEDSS, the CBRS represents the cases as a flat structure of non ordered *descriptor-value pairs*, even if this is not a state-of-the-art representation schema.

Similarity degree. This concept is used in case retrieval. The similarity between two problems P_i and P_j can be defined as complementary to their distance (D):

$$\beta(P_i, P_j) = 1 - D(P_i, P_j)$$

Two problems which are equal have the maximum similarity degree, i.e. 1, while two absolutely different problems have a minimum similarity degree, i.e. about 0.

Case retrieval. Through this process, the case with the most similar problem part with respect to the actual problem is found, searching in the case library. In OntoWEDSS, problems are represented as sets of descriptor - value pairs and the similarity between two problems is calculated as a function of the distances between the values of the descriptors (A):

$$\beta(P_i, P_j) = 1 - \frac{\sum_{k=1}^{n} D(A_k^i, A_k^j)}{n}$$

where n is the number of descriptors.

Case adaptation. When the most similar case (e.g., 29.10.199) to the actual problem (e.g, 14.12.1999) is retrieved, the CBRS adapt the solution part of the case (e.g., $WAS_{29.10.199} = 1100$). The solution is a list of actions (e.g., WAS) and each action has an associated value (e.g., 1100) and a formula. For instance:

$$WAS_{new-case} = WAS_{retrieved-case} * \frac{Water-Flow-Rate-I_{new-case}}{Water-Flow-Rate-I_{retrieved-case}}$$

$$WAS_{14.12.1999} = WAS_{29.10.199} * \frac{Water-Flow-Rate-I_{14.12.1999}}{Water-Flow-Rate-I_{29.10.199}} =$$

$$= 1100 * \frac{24000}{22000} = 1200$$

This value (1200) will be also the value of the suggested actuation, in the case of CBRS's diagnosis selection.

7.2.3 Ontology

Much has already been said about the WaWO ontology in chapter 6. We will now see more formal details on its implementation and related problems. Let us start with a description of general problems detected:

- *Undefined terms.* The construction of the WaWO ontology is an incremental process. While this process is in progress, many terms remain undefined or partially defined. The Ontolingua development environment allows the user to incrementally complete the ontology, but the OntoWEDSS system could not to function in the presence of undefined concepts.

- *Inconsistencies and conflicts*. Ontolingua ontology construction tool provides basic capabilities for avoiding conflicts and inconsistencies. If these conflicts arise during ontology manipulation after the translation to LISP, it is not possible to automatically identify the source of the problem. The OntoWEDSS system does not detect logical conflicts and their sources (i.e., sets of conflicting assumptions).

- *Completeness*. We can assess the completeness of WaWO verifying that: all referenced terms are defined, all required fields of an instance are filled, and, when a concept is exhaustively partitioned, all its instances belong to exactly one of the partition subclasses. Completeness is not only useful during the initial construction of the ontology, but is also essential for continued integrity and maintenance of the knowledge bases.

- *Semantic differences between related definitions*. While composing or updating the WaWO ontology, we are often faced with the need to identify semantic differences between two concepts (similar concepts in different Ontolingua ontologies, or the same concept in different contexts of the WaWO ontology). In OntoWEDSS there are no reasoning capabilities that can compare two concepts to identify semantic differences between them or tools for comparing different versions of an ontology. This comparison capability between concepts would be beneficial both for merging and maintenance.

With respect to the translation of WaWO from Ontolingua to LISP, we had to define or adapt a few basic routines to integrate the ontology into a standard CLOS environment. The most important procedure is a class-defining macro, which allow the generation of the hierarchical structure. A number of other functions and macros permit to keep track of defined instances.

We give now a brief overview of the main Common LISP routines:

Define-Class. A macro that expands into *defclass*, allowing an abbreviated class definition whereby all slots get accessors with the same name, *initargs* with the same name except for the colon and an *initform* if a value is supplied. It optionally allows other slot keywords (e.g., :documentation, :allocation, :type) and has a special keyword called :*Doc-String*. It also adds the mixin class[1] *Named-Object* to the list of superclasses of the class in question, automatically adding a unique *Name* slot to each instance, and creating a hash table whereby instances can be retrieved by name. This eliminates much of the bookkeeping associated with keeping track of instances in variables and allows a semantic-net like structure (where instances are stored as values of slots) to be represented in permanent code.

The simplest use of the *Define-Class* macro is that

```
(Define-Class Class (Superclasses) (Slot Val) ... )
```

expands into a *defclass* defining the class and slots, with the addition of adding *accessors* and :*initargs* with the same name as the slot name, and adding a *Name* slot/accessor by making *Named-Object* one of the superclasses. Instead of (*Slot Val*) a list (*Slot Val Slot-Specs*) can be specified, where each slot-spec is a sequence of keyword/value pairs. For instance:

```
(Define-Class Foo (Bar)
  (Slot-1 Val-1) ; <== Most common case: (Slot Value) pairs
  (Slot-2 Val-2 :Doc-String "Slot-2 string"
                :type fixnum ; A fixnum is an integer whose
                             ; value is between most-
                             ; negative-fixnum and most-
                             ; positive-fixnum inclusive.
                :allocation :class) ; Allocation-types are:
                             ; :instance and :class.
  Slot-3
  ...
  (Slot-N Val-N))
```

that expands into

[1] Sometimes in object oriented programming a class needs to inherit from two or more classes to assemble its behavior. *Mixin classes* are dedicated classes that can be used for multiple inheritance.

```
(defclass Foo (Bar Named-Object)
  ((Slot-1 :initform Val-1 :accessor Slot-1 :initarg :Slot-1)
   (Slot-2 :initform Val-1 :accessor Slot-2 :initarg :Slot-2
                           :type fixnum :allocation :class)
   (Slot-3                 :accessor Slot-3 :initarg :Slot-3)
   ...
   (Slot-N :initform Val-N :accessor Slot-N :initarg :Slot-N)))
```

with the side effect that *Slot-2 string* gets set as doc string for the generic function SLOT-2.

Alternatively, the class name (*Foo* here) can be replaced with a list of (*Class-name Class-options*), where each class-option (enclosed in parentheses) is any of the legal options for *defclass*. Thus,

```
(Define-Class (Foo (:documentation "A class called Foo")) (Bar)
  (Slot-1 Val-1)
  Slot-2)
```

expands into

```
(defclass Foo (Bar Named-Object)
  ((Slot-1 :initform Val-1 :accessor Slot-1 :initarg :Slot-1)
   (Slot-2 :accessor Slot-2 :initarg :Slot-2))
  (:documentation "A class called FOO"))
```

As noted above, making *Foo* a *Named-Object* makes a *Name* slot with a default value of *Foo-XX* (for the lowest *XX* where *Foo-XX* is not already an existing instance name). If a *Name* slot is specified explicitly, different names for each instance will be given. It is no problem to give a particular name to an *instance* when creating the object; a :*Name* initarg is created for that purpose. This name has to be a symbol. The code of *Define-Class* is:

```
(defmacro Define-Class (Class-Name Super-Class-List
                        &rest Slot-Entries)
  (if
    (atom Class-Name)
    `(defclass ,Class-Name (,@Super-Class-List Named-Object)
     ,(mapcar #'Expand-Slot-Name-Value-Pair Slot-Entries))
```

```
    `(defclass ,(first Class-Name)
       (,@Super-Class-List Named-Object)
      ,(mapcar #'Expand-Slot-Name-Value-Pair Slot-Entries)
       ,@(rest Class-Name)) )
)
```

Examples:

```
(Define-Class Thing ())

(Define-Class (Wastewater-Domain
               (:documentation
                  "The domain of wastewater treatment."))
               (Thing))

(Define-Class (Wastewater-Microbiological-Taxonomy
               (:documentation "The taxonomy of micro-organisms
                involved in wastewater treatment."))
              (Wastewater-Domain))

(Define-Class Abundance-Of-Micro-Organisms
              (Wastewater-Microbiological-Taxonomy)
              (Direct-Types 'Set-Or-Collection))
```

Def-Class. This macro is just like *Define-Class* except that the class is not automatically created as a subclass of *Named-Object*. The advantage of this is instance creation speed, which is increased by more than 10 fold. The drawback is that: (1) there is no *Name* slot, (2) it is not possible to retrieve instances by name and (3) any method (e.g. print-object and after methods on initialize-instance) that is defined to work on all custom objects in WaWO generally specializes on *Named-Object* and thus will miss this.

```
(defmacro Def-Class (Class-Name Super-Class-List
                     &rest Slot-Entries)
  (if
    (atom Class-Name)
    `(defclass ,Class-Name ,Super-Class-List
     ,(mapcar #'Expand-Slot-Name-Value-Pair Slot-Entries))
    `(defclass ,(first Class-Name) ,Super-Class-List
        Named-Object
```

```
      ,(mapcar #'Expand-Slot-Name-Value-Pair Slot-Entries)
        ,@(rest Class-Name)) )
)
```

Expand-Slot-Name-Value-Pair. A *slot* entry is either a slot name, a list of (*Slot-Name Slot-Value*) or a list of (*Slot-Name Slot-Value* <Normal CLOS Slot-Specs Keywords>). This function expands a slot entry as follows:

- Simplest cases (no keywords):
  - ```
    Slot-Name
        --> (Slot-Name :accessor Slot-Name
                       :initarg :Slot-Name)
    ```
  - ```
    (Slot-Name Value)
        --> (Slot-Name :initform Value :accessor Slot-Name
                       :initarg :Slot-Name)
    ```

- More complicated case (extra keywords):
  - *If* there is no :*Doc-String* entry in <Extra Keywords>:

    ```
    (Name Value <Extra Keywords>)
          --> (Name :initform Value
                    :accessor Name
                    :initarg :Name
                    <Extra Keywords>)
    ```
  - ```
    (Name Value :Doc-String "Test"
                <Extra Keywords>)
          --> (Name :initform Value
                    :accessor Name
                    :initarg :Name
                    <Extra Keywords>)
    ```

    As a side effect, *Test* is set as the doc string for the function *Name*.

- Also allowed is:
  - ```
    (Name) == (Name NIL)
      --> (Name :initform NIL :accessor Name
                :initarg :Name)
    ```

```
(defun Expand-Slot-Name-Value-Pair (Slot-Entry)
  (let (Slot-Name Slot-Value Extra-Keywords Doc-String)
    (cond
      ((listp Slot-Entry)
       (setq Slot-Name (first Slot-Entry)
         Slot-Value (second Slot-Entry)
         Extra-Keywords (rest (rest Slot-Entry))
         Doc-String (getf Extra-Keywords :Doc-String))
       (when Doc-String
     (remf Extra-Keywords :Doc-String)
     (setf (documentation Slot-Name 'function) Doc-String))
       (append
     (list Slot-Name
           :accessor Slot-Name
           :initform Slot-Value
           :initarg (Add-Colon Slot-Name))
     Extra-Keywords) )
      (t
       (setq Slot-Name Slot-Entry)
       (list Slot-Name
         :accessor Slot-Name
         :initarg (Add-Colon Slot-Name)) ) )
))
```

Add-Colon. Given '*Foo* or "*Foo*" it returns :*FOO*. This has the same effect as

(read-from-string (concatenate 'string ":" (string *Argument*)))

but it does not have to invoke the LISP reader. There can be a problem in calling the second one of these methods *interactively* from the LISP listener on the *Symbolics*, as the *Symbolics* often puts font characters into strings. It can be necessary to do (Add-Colon (user::string-thin "Foo")). This is not a concern in *functions* that call *Add-Colon*, however.

```
(defmethod Add-Colon ((Sym symbol))
  (if (keywordp Sym)
      Sym
      (intern (symbol-name Sym) :keyword)))

(defmethod Add-Colon ((Str string))
  (intern (string-upcase Str) :keyword))
```

Named-Object. A mixin class that gets added to the superclass list of all classes defined with *Define-Class*. It adds a slot called *Name* and the associated reader. The default value of this slot will be :*Foo-XX*, where *Foo* is the class name of the instance being created, and *XX* is the lowest natural number whereby :*Foo-XX* does not already name an instance. All instances of *Named-Objects* get recorded in a hash table with the name as a key. There is deliberately no (setf *Name*) operator. This can be added to allow renaming, as long as the hash table is appropriately updated.

```
(defclass Named-Object ()
    ((Name :initform NIL :reader Name :initarg :Name))
  (:documentation
    "Class of objects to which all objects defined with
    Define-Class belong. It provides a 'Name' slot,
    which is used in the function Get-Instance, that
    returns the instance object with a given name, and
    in the print-object method to put the name in the
    printed representation. Named-Object is also the
    class to use to specialize a method on all objects.") )
```

CLOS-Instance-Name-Table This records the names of *all CLOS objects* that have name slots. Since it does not use :test #'equal, it will not work for instances that have strings or lists as their names. The Ontolingua convention is that it is not possible to have two distinguishable classes or instances whose names have the same symbol-name, even in different ontologies of the Ontolingua library. This was desired in Ontolingua so that object names can be looked up and accessed from multiple ontologies, but is a limitation that users should be aware of. Note that *Remove-Instance* needs to know how to remove entries both from this table and the following one.

```
(defvar *CLOS-Instance-Name-Table* (make-hash-table)
  "A hash table associating object names with the objects
   themselves.")
```

***CLOS-Class-Name-Table*.** Every time a CLOS instance is created, it is added to the list of instances in this table that are associated with the class name. Note that *Remove-Instance* needs to know how to remove entries both from this table and the preceding one.

```
(defvar *CLOS-Class-Name-Table* (make-hash-table)
  "A hash table associating class names with the direct
  instances of that class 'Instances' or
  'Instance-Names' can be used to get all instances of
  a class")
```

***CLOS-Class-Name-Counters*.** Every time an instance is created that does not have an explicit name, then a counter associated with its class is used to get *CLASS-NAME-N* as the name and then the counter is incremented.

```
(defvar *CLOS-Class-Name-Counters* (make-hash-table)
  "A hash table associating a class name with an integer.
  This integer is the next one that will be used for
  CLASS-NAME-N when providing a name for an instance.")
```

Make-instance. We use the ANSI Common LISP (Steele 1990) generic function *make-instance*, which creates and returns a new instance of the given class. The initialization of a new instance consists of several distinct steps, including the following: combining the explicitly supplied initialization arguments with default values for the unsupplied initialization arguments, checking the validity of the initialization arguments, allocating storage for the instance, filling slots with values, and executing *user-supplied methods* that perform additional initialization. Each step of make-instance is implemented by a generic function to provide a mechanism for customizing that step. Example:

```
> (make-instance 'ABUNDANCE-OF-MICRO-ORGANISMS)
#<ABUNDANCE-OF-MICRO-ORGANISMS-4
    (an ABUNDANCE-OF-MICRO-ORGANISMS)>
```

Initialize-instance. It is called by *make-instance* to initialize a newly created instance. We define this method for *initialize-instance* to specify actions to be taken when an instance is initialized. This is an *after* method, it will be run after the system-supplied primary method for initialization and therefore will not interfere with the default behavior of *initialize-instance*. Any instance that is created will get a name based on its class (unless it has an explicit name) and will be recorded in the hash table.

```
(defmethod initialize-instance :after ((Obj Named-Object)
                                       &rest Extra-Args)
  (declare (ignore Extra-Args))
  (let ((Name (Name Obj))
    Previous-Instance)
    (cond
      ((and Name (not (keywordp Name)))
       (setf Name (Add-Colon Name)))
      ((null Name)
       (setf Name (Instance-Name (class-name
                                  (class-of Obj))))))
    (setf (slot-value Obj 'Name) Name)
    (setq Previous-Instance (Get-Instance Name))
    (when Previous-Instance
      (format t "~%Replacing ~S with ~S since they have the
                same name."
            Previous-Instance Obj)
      (Remove-Instance Previous-Instance))
    (setf (gethash (Name Obj) *CLOS-Instance-Name-Table*)
          Obj)
    (push Obj (gethash (Instance-Class Obj)
                *CLOS-Class-Name-Table*))
))
```

Print-object. If an instance has a name slot, *Print-object* uses it in the printed representation. E.g., if *Name* = *Foo-3* (or :*Foo-3*) and class is *FOO*, the printed representation is:

#<FOO-3 (a FOO)>.

```
(defmethod print-object ((Obj Named-Object) Stream)
  (let ((Name (symbol-name (Name Obj)))
    (Class (class-name (class-of Obj))))
    (format Stream "#<~A (~A ~S)>" Name
            (Indefinite-Article Class) Class) ))

(defun Indefinite-Article (String)
```

```
  "Returns 'a' or 'an' depending on whether or not String
  begins with a A, E, I or O."
  (case (aref (string-capitalize String :end 1) 0)
    ((#\A #\E #\I #\O) "an")
    (otherwise          "a")))
```

Name. Reader method created automatically for all *Named-Objects*, i.e. everything created via *Define-Class*. If *Name* is called on an object that is neither a *Named-Object* nor has an explicitly defined accessor *Name*, the user gets a warning message.

```
(defmethod Name ((Obj Standard-Object))
  (format t "~%~S is not a Named-Object and has no accessor
   'Name'." Obj)
  (format t "~%Note that using 'Define-Class' automatically
   makes the class~%~ a subclass of Named-Object.")
)
```

Instance-Name. Given '*Foo* returns :*FOO-1* or :*FOO-2* or in general :*FOO-N* for the smallest value of N such that :*FOO-N* has never been an existing instance name. This is better than using ANSI LISP *gentemp* since *gentemp* does not necessarily number independently. I.e.,

```
(gentemp "ABUNDANCE-OF-MICRO-ORGANISMS-" :keyword)
  --> :ABUNDANCE-OF-MICRO-ORGANISMS-6
```

but

```
(gentemp "WASTEWATER-MICROBIOLOGICAL-TAXONOMY-" :keyword)
  --> :WASTEWATER-MICROBIOLOGICAL-TAXONOMY-7,
      not :WASTEWATER-MICROBIOLOGICAL-TAXONOMY-1.
```

Here, we prefer each class to have its own separate numbering.

```
(defun Instance-Name (Class-Name)
  "Given a symbol such as 'Foo', returns ':Foo-XX' for the
  next natural number XX for which :Sub-XX is not already
  an existing instance name."
  (let* ((N (incf (gethash Class-Name
                  *CLOS-Class-Name-Counters* 0)))
```

```
     (Name (intern (concatenate 'string
                     (symbol-name Class-Name)
                     "-"
                     (princ-to-string N))
                :Keyword)))
    (if
      (Get-Instance Name)
      (Instance-Name Class-Name)
      Name)
))
```

Get-Instance. A method that takes an instance name as an argument, and returns the instance with that name. Defined macro characters exist for this method, such that:

{*Foo*} == (Get-Instance :*Foo*) and

[*Foo*] == (Get-Instance *Foo*), so that for instance

(Depth {*WWTP-2*}) == (Depth (Get-Instance '*WWTP-2*))

Copy-Instance. It takes an instance and copies all slot values to another. The result is as *both* instances were made via *Define-Class*: they have identical slot names.

Assign-Slot-Value. Given a quoted *Instance* name, *Slot* name and *Value*, it does (setf (*Slot Instance*) *Value*)

Remove-Instance. It takes the name of an instance and removes the corresponding entry in the hash table.

Remove-Instances. It removes (in the sense noted above) all instances of a specified class.

Direct-Instances. It takes a class name and returns all instances of *Named-Objects* that are directly (no intervening subclasses) in that class.

Instances. It takes a class name and returns, unsorted, all instance of *Named-Objects* which are directly or indirectly in that class.

Instance-Names. It returns the *names* (not the objects) of all instances of *Named-Objects* that are directly or indirectly in the specified class. The names are sorted alphabetically if the :*Sort-p* flag is set.

All-Instances. It returns all instances of *Named-Objects*.

All-Instance-Names. It returns the *names* (not the objects) of all instances of *Named-Objects*. The names are sorted alphabetically if the :*Sort-p* flag is set.

Slot-Names. Given an instance or a class name, it returns a list of all slots.

Direct-Slot-Names. Given an instance or a class name, it returns a list of all *directly* defined slots. I.e., inherited slots are not included.

Has-Reader-p. Given an instance and a slot name, it determines if there is a reader method with the same name as the slot, as *Define-Class* would make automatically.

Instance-Class. Given an instance name or an instance object, it returns a symbol that is the immediate class name. I.e., given *Appearance-Floc-1* returns *Appearance-Floc*. If the argument is neither an instance nor an instance name, this returns *NIL*.

Subclasses. Given a class name, it returns the names of the direct subclasses. It returns *NIL* if there are no subclasses *or* if the supplied symbol names no class. The names returned are sorted alphabetically if the :*Sort-p* flag is set.

Internal-Address-String. A non-standard way to get the address of an object. *Internal-Address-String* returns it in a string for use by a specialized print-object. It is *not* portable to other implementations.

7.3 Decision-support layer

The *decision support* (DS) layer helps the manager of a WWTP to make decisions about the supervision and control of the plant. The decision support layer obtains information from the diagnosis layer and processes it in order to support problem-specific decision-making. The DS layer provides clear and useful answers to the set of complex questions represented by a new situation of the plant, to which an uneducated operator does not have the capacity to answer herself. The information obtained from this layer is then used to arrive at operational decisions.

The DS layer support different types of decisions to be made such as one-of-a-kind decisions or repetitive decisions. It can be used to support highly structured problems as well as unstructured problems. With access to more precise and accurate information, WWTP users can make better supervisory decisions, implement the right actions, and target the *right* cause of the problem. Even if, in general, the DS layer could be used on several levels, in OntoWEDSS it is brought into play mainly at the operational level. For instance, in the case that CBRS's diagnostics is passed to the decision support layer, the description of the associated action (e.g., "Check out *Food-To-Micro-Organism-Ratio*", "Remove aeration-tank and clarifier foam" "Reduce *FlowRate-WAS*" or "Use minimum recycled-activated-sludge flow rate to facilitate good compacting") and the result of calculating the action's formula are exactly what OntoWEDSS will suggest to the user.

The decision support layer consists of: (1) an engine to access the diagnosis-integrator module and extract, if available, the information about actuation, (2)

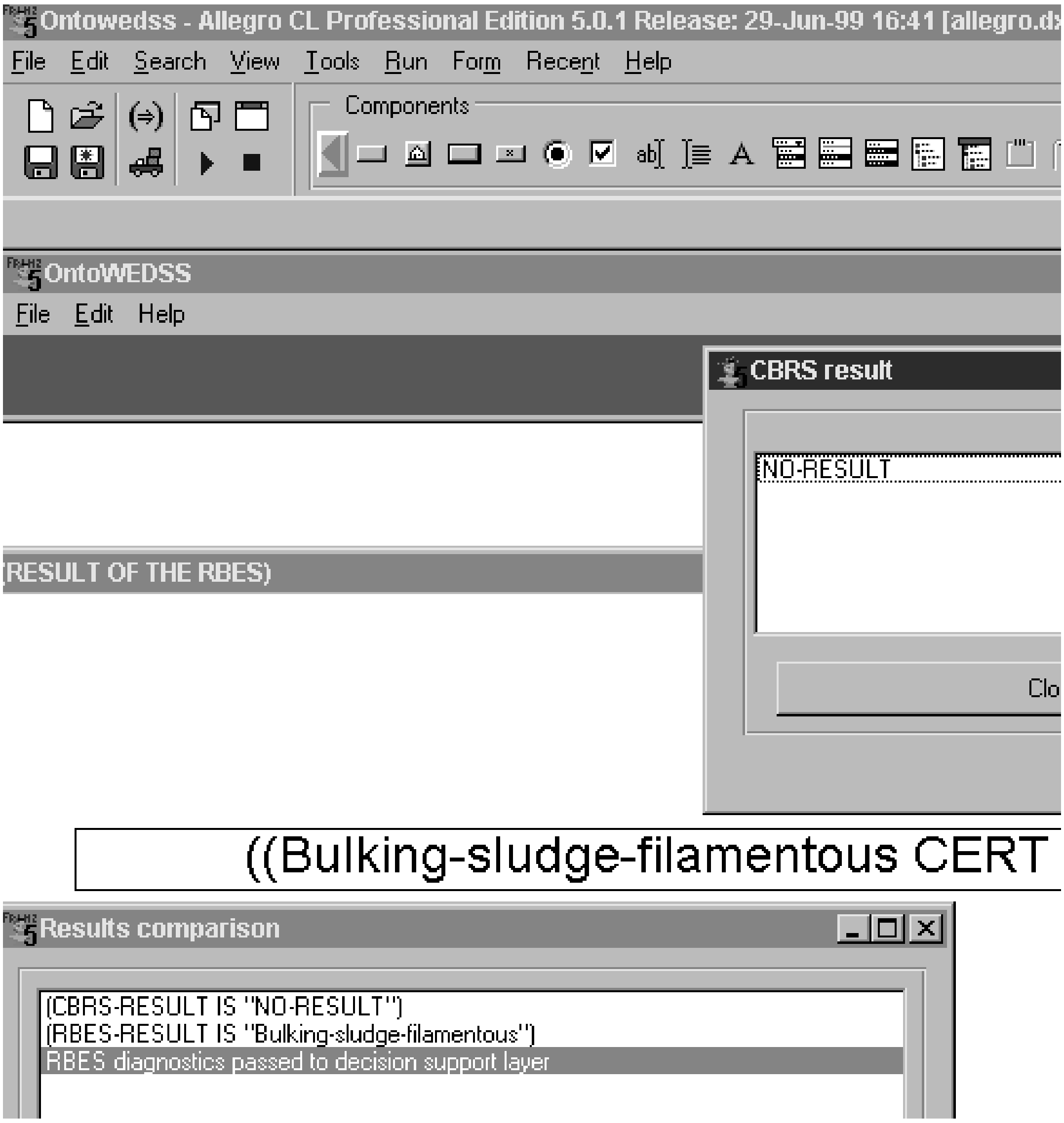

Figure 7.4: Comparison of CBRS and RBES results.

an explanation module to inspect the analysis process of the case-based reasoning and the rule-based reasoning and (3) a user interface. With the incorporation of a simulation module, users will possess the ability not only to understand the current WWTP's situation and operate according to it, but also to predict future WWTP's trends. The interface enables the end users to have access to integrated and analytical information rather than raw data (see Figure 7.4.

7.4 How to use OntoWEDSS

The use of OntoWEDSS can conceptually be subdivided in two parts: domain modelling and execution. We will describe all the functionalities of OntoWEDSS through working examples reflecting real situations, which have been slightly simplified.

7.4.1 Domain modelling

The domain-modelling part of OntoWEDSS facilitates search space handling, result presentation and domain organization.

To start OntoWEDSS, load (double clicking) the *ontowedss-project* LISP file in the OntoWEDSS folder, then *run* the project. The red main window appears.

Domain definition and modification in the CBRS.

1. (In OntoWEDSS) File > Case-based reasoning system > Open.
    (a) (In the CBRS) User: *superuser*.
    (b) Password: *srbc*.
2. Domain > Create domain.
3. Domain name: *Wastewater*.
    (a) i. Descriptor name: *Water-Flow-Rate-I*.
        ii. Descriptor type: quantitative.
        iii. Weight: 0.6.
        iv. Number of intervals: 3.
            A. Interval beginning: 0.
            B. Interval end: 19000.
            C. Interval name: *Low*.
            A. Interval beginning: 19000.

        B. Interval end: 24000.

        C. Interval name: *Normal*.

        A. Interval beginning: 24000.

        B. Interval end: 60000.

        C. Interval name: *High*.

    (b) And so on for all descriptors of the domain

As an alternative to creation from scratch, a domain can be loaded, visualized in the *Information about the domain* window and then, if necessary, edited:

1. Domain > Load domain.
2. Path of file: OntoWEDSS\Domains.
3. File's name: *granollers*. (This domain, which we use throughout the section, includes 21 descriptors.)

Domain definition and modification in the RBES.

1. (In OntoWEDSS) File > Rule-based expert system > Open.
2. (In the RBES) KB management > New KB.
    (a) Name: *granollers*.
3. ES design > Type.
    (a) ES design > Type > New type.
    (b) Type's name: *Filam-Dominant-AT*.
    (c) Add new value: *Microthrix-Parvicella*.
    (d) Add new value: *Gordona*.
    (e) Add new value: *Thiothrix*.

    (f) ... and so on for all the values of *Filam-Dominant-AT*.

    (a) ES design > Type > New type.
    (b) Type's name: *Diagnosis*.
    (c) Add new value: *Electrical-Blackout*.
    (d) Add new value: *Foaming-Sludge*.
    (e) Add new value: *Bulking-Sludge-Filamentous*.
    (f) ... and so on for all the values of *Diagnosis*.

4. ES design > Data.
    (a) ES design > Data > New data.
        i. Name: *Water-Flow-Rate-I*.
        ii. Type: numerical.
        iii. Value: no value.
        iv. Interaction possibility: yes.
        v. Question: Water-Flow-Rate-I?
    (b) ES design > Data > New data.
        i. Name: *TSS-E*.
        ii. Type: numerical.
        iii. Value: no value.
        iv. Interaction possibility: yes.
        v. Question: TSS-E?
    (c) ES design > Data > New data.
        i. Name: *SVI-AT*.
        ii. Type: numerical.

        iii. Value: no value.
        iv. Interaction possibility: yes.
        v. Question: SVI-AT?

    (d) ES design > Data > New data.

        i. Name: *Bulking-sludge-filamentous*.
        ii. Type: boolean.
        iii. Value: no value.
        iv. Interaction possibility: no.

    (e) ... and so on for all data.

5. ES design > Rule.

    (a) ES design > Rules > New rule.

        i. Name: *bsf-1* (first rule for bulking-sludge-filamentous).
        ii. Module: no module.
        iii. Add invocation condition.

            A. Data 1: *TSS-E*.
            B. Operator: >.
            C. Numerical constant: 21.

        iv. Rule conclusion.
        v. Data: *pre-alarm-filamentous-bulking*.
        vi. Operator: is.
        vii. Non numerical value: true (select it even if it is the default value).

    (b) ES design > Rules > New rule.

        i. Name: *bsf-2*.
        ii. Module: no module.

iii. Add invocation condition.

   A. Data 1: *pre-alarm-filamentous-bulking*.

   B. Operator: is.

   C. Non numerical constant: true.

iv. Add invocation condition.

   A. Data 1: *SVI-AT*.

   B. Operator: >.

   C. Numerical constant: 140.

v. Add invocation condition.

   A. Negation: yes.

   B. Data 1: *filam-dominant-at*.

   C. Operator: is.

   D. Non numerical constant: *none*.

vi. Rule conclusion.

vii. Data: *Bulking-Sludge-Filamentous*.

viii. Operator: is.

ix. Numerical value: true.

x. ... and so on for all rules.

Domain definition in WaWO. WaWO is the foundation of the modelling of the domain. It serves search space handling in the following way:

- Expanding the search: querying with similar concepts (using the boolean *or*);
- Reducing the search: querying with more specific concepts;
- Searching cross-lingually: expanding the search using available translations of the terms (in the case that non-standard categories are used).

(In OntoWEDSS) File > Ontology > Open.

In this way the whole WaWO is loaded. It is not possible to edit the ontology directly from the user interface. It is instead possible to do it using the routines described in §7.2.3.

7.4.2 Execution

Execution of the CBRS.

1. Domain > Load domain

    (a) Path of file: OntoWEDSS\Domains.

    (b) File's name: *granollers*.

2. Cases > Load cases' list learning all

    (a) Path of file: OntoWEDSS\Cases.

    (b) File's name: *granollers-all-185*.

3. Cases > Introducing a new problem
   We introduce the following problem: (”Day” 5 ”Month” 11 ”Year” 2000 ”Hour” ”01-00pm” ”Water-Flow-Rate-I” 22166 ”TSS-I” 276.00 ”COD-I” 739.00 ”TKN” ” ” ”TSS-P” 88.00 ”COD-P” 556.00 ”MLSS-AT” 2660.0 ”SVI-AT” 139.1 ”SRT-AT” 0.00 ”F-M-AT” 0.00 ”Filam-Dominant-AT” ”?” ”DO-AT-1” 2.45 ”DO-AT-2” 2.44 ”TSS-E” 27.00 ”COD-E” 96.00 ”TN-E” ” ” ”Diagnosis” ”?” ”WAS” - ”RAS” - ”Air-Flow-1” - ”Air-Flow-2” -)

4. The CBRS try to retrieve a similar case and the result is that no case has been retrieved with a similarity greater then β. The 5 ”most similar” cases are shown to the user anyway in a pop-up window.

Execution of the RBES.

1. Expert system execution > Load > granollers

2. Expert system execution > Execution > Start execution > Start execution

   (a) "filam-dominant-at?": *Microthrix-Parvicella*

   (b) "TSS-E?": 30

   (c) "SVI-AT?" 150

Conclusions appear in the *Deduced conclusions* window (see Figure § 7.5 on the facing page for an example):

(*Bulking-sludge-filamentous* true 1.0)

and the trace of the reasoning appears in the *Trace* window. When the *Back* button is pressed, a new window appears with the final result of the rule-based expert system.

Execution of WaWO. After the execution of the CBRS and the RBES, OntoWEDSS automatically executes and compares the results as described in §5.4.3 (see Figure §7.4).

7.5 Discussion on contributions

In this section we discuss the contributions of OntoWEDSS regarding several issues strictly related to implementation.

- *Language.* LISP was chosen as the implementation language. These days, most AI programs are written in LISP. OntoWEDSS is no exception.

- *Portability among platforms.* OntoWEDSS was developed in Allegro Common Lisp v.5. This is a commercial, object oriented development-environment implemented on the Common LISP language, which allowed the construction of

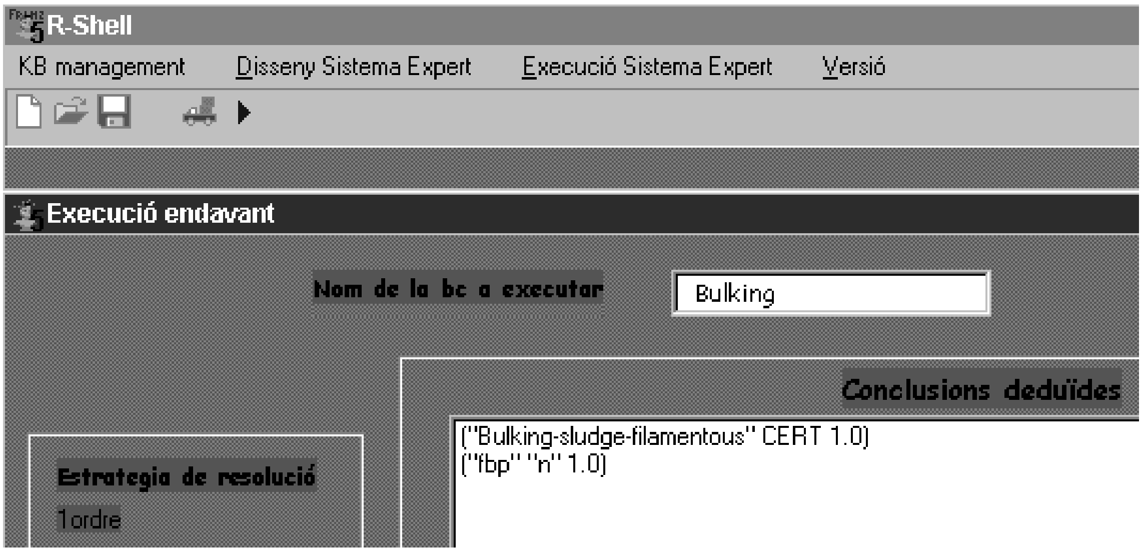

Figure 7.5: Window for the result of the rule-based expert system.

an application that can run on Windows, Macintosh and UNIX platforms.

- *Comparison with other systems.* With respect to the RBES, other development environments for expert systems with similar characteristics are: BABYLON (Christaller *et al.* 1992), implemented in Common LISP; CLIPS (Giarratano and Riley 1993), implemented in ANSI C; MILORD II (Puyol and Sierra 1997), developed in Common Lisp (the interpreter) and in C (the compiler). With respect to the CBRS, other similar shells which, in addition, use decision trees to index cases are: Kaidara's engine[2], Easy Reasoner[3], Induce-It[4], ReMind[5] With respect to WaWO, there are no other ontologies for the wastewater domain, as noted in §2.4.7, where, on the other hand, general development environment for ontologies are depicted.

[2]http://www.acknosoft.com/

[3]Easy Reasoner is based on Eclipse, http://www.haley.com/

[4]http://www.inductive.com/

[5]ReMind, produced by Cognitive Systems Inc., was developed with support from the US DARPA military programme. It was originally developed for the Macintosh and has since been ported to MS Windows and UNIX platforms. It is available as a C library for embedding in other applications and as an interactive development environment. In 1996, Cognitive Systems ceased trading; however, ReMind Version 2.0 is under development at the Navy Center for Applied Research in Artificial Intelligence in Washington DC. It is not clear though when the new version will be released and who may retail it. ReMind 1.1 is still widely used and has been a very influential CBR tool.

- *Case retrieval in the CBRS.* To retrieve the most similar case, two criteria are used: nearest cases and, secondarily, cases with the best outcome.

Chapter 8

Evaluation

> Tirada en el campo estaba desde hacía tiempo una Flauta que ya nadie tocaba, hasta que un día un Burro que paseaba por ahí resopló fuerte sobre ella haciéndola producir el sonido más dulce de su vida, es decir, de la vida del Burro y de la Flauta.
> Incapaces de comprender lo que había pasado, pues la racionalidad no era su fuerte y ambos creían en la racionalidad, se separaron presurosos, avergonzados de lo mejor que el uno y el otro habían hecho durante su triste existencia.
>
> *Augusto Monterroso*

This chapter deals with a focussed evaluation of the OntoWEDSS methodology as a supervisory procedure for WWTP management and includes the individual evaluation of each piece of technology associated to OntoWEDSS's hybrid architecture. This evaluation is focussed on the most representative problematic situations that it is possible to come upon in wastewater treatment. Here, we present the results relative to the presence of bulking sludge due to filamentous micro-organisms and in one of the following sections we will justify why we chose this particular circumstance. The objective is a mixed evaluation of quantitative and qualitative aspects of the various paradigms and of the whole system when they react and provide an answer to specific problems. The procedure described can also be extended to a generic situation.

Evaluation in environmental domains in general and particularly in wastewater treatment is problematic and complex for a number of reasons:

- *Lack of benchmarks.* There is a lack of benchmarks that work, with regards to wastewater management. Now that so many WWTPs are in operation, an accurate benchmark is critically needed. This situation does not allow an accurate quantitative measure of management improvement.

- *High number of descriptors.* To model a complex domain a high number of descriptors is required. OntoWEDSS use more descriptors than previous similar systems, even if experts helped us to select just 20 descriptors among all 170 available ones. The chosen descriptors are the *most* relevant in experts' practice and experience.

- *High percentage of missing values.*In real world environmental applications we are usually faced with a high number of instances with missing descriptor's values. These instances are often not suitable to be correctly labelled.

- *Multiple labels.* In wastewater treatment domain it is possible to assign more than one *Diagnosis* label to a state of the plant (e.g., *Bulking-Sludge*, *Underloading* and *Rising-Sludge*), ordered according to importance. This situation makes the evaluation of diagnostics more difficult. We chose to work with just one label per instance to ease the validation process and this degrades in part the CBR's performance.

8.1 Evaluation design

For evaluation we used an initial set of 790 data corresponding to more than two years (from 1.4.1998 to 9.10.2000) of real-operation of a treatment plant located in Catalonia. Each instance corresponds to one day and contains the mean values of 21 descriptors for that day. The set of all these values, except four of them (diagnosis and actuation), represents the state of the plant for that day. This state is labelled as *Diagnosis*.

Due to the high percentage of missing values (more than 50% of descriptors), we discarded part of the data and keep instances from 2.12.1998 to 29.4.1999 and from 5.7.1999 to 9.10.2000, which nonetheless present about 20% of missing values. This filtering assured a better quality in the training set. Then, we selected for the experiments only the data for which a diagnosis label exists and called *G-186* the resulting set of 186 labelled days:

```
(("Day" 2 "Month" 12 "Year" 1998 "Hour" "10-0am"
"Water-Flow-Rate-I" 26850.00 "TSS-I" 214.00 "COD-I" 761.00 "TKN"
nil "TSS-P" 118.00 "COD-P" 437.00 "MLSS-AT" 2591.0 "SVI-AT" 88.8
"SRT-AT" 8.11 "F-M-AT" 0.44 "Filam-Dominant-AT" "Gordona"
"DO-AT-1" nil "DO-AT-2" nil "TSS-E" 23.00 "COD-E" 124.00 "TN-E"
nil "Diagnosis" "normal-operation" "WAS" nil "RAS" 121.48
"Air-Flow-1" nil "Air-Flow-2" nil)

("Day" 5 "Month" 12 "Year" 1998 "Hour" "10-0am"
"Water-Flow-Rate-I" 23909.00 "TSS-I" 216.00 "COD-I" 555.00 "TKN"
nil "TSS-P" 92.00 "COD-P" 407.00 "MLSS-AT" 2836.0 "SVI-AT" 102.3
"SRT-AT" 8.79 "F-M-AT" 0.30 "Filam-Dominant-AT" "Gordona"
"DO-AT-1" nil "DO-AT-2" nil "TSS-E" 11.00 "COD-E" 101.00 "TN-E"
nil "Diagnosis" "normal-operation" "WAS" nil "RAS" 92.71
"Air-Flow-1" nil "Air-Flow-2" nil)

("Day" 7 "Month" 12 "Year" 1998 "Hour" "10-0am"
"Water-Flow-Rate-I" 22333.00 "TSS-I" 210.00 "COD-I" 606.00 "TKN"
nil "TSS-P" 84.00 "COD-P" 362.00 "MLSS-AT" 3236.0 "SVI-AT" 92.7
"SRT-AT" 7.53 "F-M-AT" 0.22 "Filam-Dominant-AT" "Gordona"
"DO-AT-1" nil "DO-AT-2" nil "TSS-E" 9.00 "COD-E" 115.00 "TN-E" nil
"Diagnosis" "normal-operation" "WAS" nil "RAS" 140.51 "Air-Flow-1"
nil "Air-Flow-2" nil)
```

```
...

("Day" 5 "Month" 10 "Year" 2000 "Hour" "10-0am"
"Water-Flow-Rate-I" 22166 "TSS-I" 276.00 "COD-I" 739.00 "TKN" nil
"TSS-P" 88.00 "COD-P" 556.00 "MLSS-AT" 2660.0 "SVI-AT" 139.1
"SRT-AT" 0.00 "F-M-AT" 0.00 "Filam-Dominant-AT" nil "DO-AT-1" 2.45
"DO-AT-2" 2.44 "TSS-E" 27.00 "COD-E" 96.00 "TN-E" nil "Diagnosis"
"organic-overloading" "WAS" 0.0 "RAS" 37.3 "Air-Flow-1" 63364
"Air-Flow-2" 66902))
```

To evaluate the performance of the *CBRS*, we performed a 3-fold cross-validation (subset of 62 days). The test has been carried out with the data of the testing subset which were labelled as *Bulking-Sludge-Filamentous*. Obviously the *Diagnosis* descriptor was not used for case retrieval. Finally, we calculated the mean percentage success and the standard deviation of the CBRS in retrieving a case with a *Bulking-Sludge-Filamentous* diagnosis.

To evaluate the performance of the *RBES*, we separated all data of *G-186* which were labelled as *Bulking-Sludge-Filamentous* in three subsets, according to CBRS's testing-set partition. Then we apply the RBES to them and observe the percentage success and the standard deviation of the rule system in detecting the bulking situation.

The results of CBRS and RBES have been then intersected to find the cases in which no positive diagnosis was produced and we analyzed the behavior of *WaWO* with respect to these cases. The percentage of success in those impasse situations is an evaluation of the improvement in the diagnosis of *Bulking-Sludge-Filamentous* of the overall OntoWEDSS system, with respect to using only CBRS and RBES together. Thus, we could get an insight of the improvement obtained by OntoWEDSS with respect to other similar decision-support systems.

8.2 Why the bulking-sludge state?

The activated sludge process of WWTPs relies on the operation of two main units: the biological reactor and the secondary settler. We chose to focus the attention on *Bulking-Sludge-Filamentous* episodes because of its significance and consequences within this activated sludge process.

The phenomenon of bulking sludge can be described as the *excessive growth of filamentous bacteria delaying sludge settling* in the secondary clarifier of the plant and resulting in poor effluent water quality. This circumstance occurs when the biomass is strongly colonized by long bacteria, whose filaments keep the flocs apart, interfering in this way with compacting and settling of activated sludge. This excessive increase of filamentous organisms is in part the result of the implementation of methods for the removal of nutrients. Various review studies have established that, among the filamentous-organisms types observed ubiquitously in activated sludge, approximately 10 types account for the great majority of bulking episodes. Thus, the microscopic identification of these types, their abundance, condition and growth forms of the filaments provides precious information about the nature and the causes of bulking problems. There are several factors influencing the occurrence of filamentous types (e.g., *Sludge-Residence-Time*, *Food-To-Micro-Organism-Ratio*, *Dissolved-Oxygen*, concentration of nutrients or pH) and the identification of the *Dominant-Filamentous-Bacteria* is an important key to discover the main responsible factor for bulking.

The implementation of a better control of bulking episodes would critically improve the operation of a plant when this problem alters its performance. This certainly involves also the analysis of on-line signals, analytical determinations and microbiological examinations. A lot of research effort has been dedicated to understand and reduce bulking sludge episodes (Jenkins *et al.* 1993; Wanner 1994) but, though advances have been made, it is still a very incisive problem in biological wastewater treatment. One of the hardest difficulties is the automatic diagnosis of the *Dominant-Filamentous-Bacteria*. Once the cause of bulking is identified, a specific solution can be proposed to restore the correct process.

8.3 Evaluation of the CBRS

The initial *G-186* data set has been randomly partitioned into 3 mutually exclusive partitions. On each of the runs the CBRS algorithm learned, using 2 of these partitions as case library, and has been evaluated on the instances of the remaining partition which are labelled as *Bulking-Sludge-Filamentous*. The result of the cross validation is shown in Table 8.1.

The mean success rate has been obtained as an average of the success on each of the three iterations. To compare CBRS and RBES algorithms we tested all of them on the same iterations and compared the estimated successes. However this comparison may not be sufficiently reliable and we complemented this average success by adding the standard deviation and the semi-difference between maximum and minimum values over all iterations:

Mean successful outcomes: 27%

Standard deviation: 8%

Semi-difference between maximum and minimum values: 8%

Table 8.1: CBRS's evaluation.

Experiment	**Testing-set data**	**Correct case-retrieval**
G-1	8	25%
G-2	10	20%
G-3	11	36%

8.4 Evaluation of the RBES

Table 8.2: RBES's evaluation.

Experiment	Number of data	Correct classification
G-1	8	38%
G-2	10	50%
G-3	11	73%

We used for RBES the testing sets of CBRS, which include all instance labelled as *Bulking-Sludge-Filamentous*. The result of the application of the rule system on the three sets is shown in Table 8.2.

The mean success rate is obtained as an average of the success on each of the three sets. We complement this average successes with the standard deviation and the semi-difference between maximum and minimum values over the three sets:

Mean successful outcomes: 54%

Standard deviation: 18%

Semi-difference between maximum and minimum values: 18%

The advantage of using the CBRS plus the RBES is clarified in Table 8.3, where successful diagnosis coming from either one of the two system is considered.

Table 8.3: Evaluation of CBRS plus RBES.

Experiment	Number of data	Correct classification
G-1	8	63%
G-2	10	60%
G-3	11	73%

Mean successful outcomes: 65%

Standard deviation: 7%

Semi-difference between maximum and minimum values: 7%

8.5 Evaluation of the complete system

We now scrutinize the results obtained from the CBRS and RBES on the 19 instances labelled as *Bulking-Sludge-Filamentous* and then we show how the ontology can improve these results. In the field of ontologies an important drawback is the lack of accurate evaluation techniques, therefore we carry out a case by case qualitative evaluation.

G-1 set results.

- ```
  ("Day" 5 "Month" 12 "Year" 2000 "Hour" "10-0am"
  Water-Flow-Rate-I" 19678 "TSS-I" 248.0 "COD-I" 649.0 "TKN" NIL
  "TSS-P" 104.0 "COD-P" NIL "MLSS-AT" NIL "SVI-AT" NIL "SRT-AT"
  0.0 "F-M-AT" 0.0 "Filam-Dominant-AT" NIL "DO-AT-1" 1.5
  "DO-AT-2" 1.51 "TSS-E" 15.0 "COD-E" 92.0 "TN-E" NIL "Diagnosis"
  "bulking-sludge-filamentous" "WAS" 1037.4 "RAS" 100.1
  "Air-Flow-1" 81942 "Air-Flow-2" 77686)
  ```

  ↪ Identified by CBRS.

- ```
  ("Day" 24 "Month" 9 "Year" 2000 "Hour" "10-0am"
  "Water-Flow-Rate-I" 15613 "TSS-I" 284.0 "COD-I" 688.0 "TKN"
  NIL "TSS-P" 156.0 "COD-P" NIL "MLSS-AT" 3260.0 "SVI-AT"
  214.7 "SRT-AT" 12.19 "F-M-AT" 0.0 "Filam-Dominant-AT" NIL
  "DO-AT-1" 1.91 "DO-AT-2" 2.14 "TSS-E" 25.0 "COD-E" 108.0
  "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS"
  415.3 "RAS" 77.2 "Air-Flow-1" 81112 "Air-Flow-2" 67501)
  ```

  ↪ Identified by RBES.

- ```
  ("Day" 25 "Month" 10 "Year" 1999 "Hour" "10-0am"
  "Water-Flow-Rate-I" 23500 "TSS-I" 132.0 "COD-I" 686.0 "TKN"
  120.0 "TSS-P" 72.0 "COD-P" 498.0 "MLSS-AT" NIL "SVI-AT" NIL
  "SRT-AT" NIL "F-M-AT" NIL "Filam-Dominant-AT" "Microthrix"
  "DO-AT-1" 3.71 "DO-AT-2" 2.51 "TSS-E" 14.0 "COD-E" 75.0
  "TN-E" 52.3 "Diagnosis" "bulking-sludge-filamentous" "WAS"
  266.0 "RAS" 92.0 "Air-Flow-1" NIL "Air-Flow-2" NIL)
  ```

  ↪ Impasse situation.

- ```
  ("Day" 23 "Month" 5 "Year" 2000 "Hour" "10-0am"
  "Water-Flow-Rate-I" 27472 "TSS-I" 216.0 "COD-I" 635.0 "TKN"
  ```

64.0 "TSS-P" 92.0 "COD-P" 538.0 "MLSS-AT" 3260.0 "SVI-AT" 92.0 "SRT-AT" 6.45 "F-M-AT" 0.23 "Filam-Dominant-AT" "Microthrix" "DO-AT-1" 2.0 "DO-AT-2" 2.16 "TSS-E" 25.0 "COD-E" 85.0 "TN-E" 53.4 "Diagnosis" "bulking-sludge-filamentous" "WAS" 1613.5 "RAS" 100.5 "Air-Flow-1" 201600 "Air-Flow-2" 183000)

↪ Impasse situation.

- ("Day" 16 "Month" 5 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 21849 "TSS-I" 112.0 "COD-I" 523.0 "TKN" NIL "TSS-P" 64.0 "COD-P" 472.0 "MLSS-AT" 3340.0 "SVI-AT" 101.8 "SRT-AT" 5.39 "F-M-AT" 0.0 "Filam-Dominant-AT" NIL "DO-AT-1" 1.49 "DO-AT-2" 1.5 "TSS-E" 14.0 "COD-E" 86.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 1010.1 "RAS" 100.4 "Air-Flow-1" 101000 "Air-Flow-2" 95819)

↪ Impasse situation.

- ("Day" 7 "Month" 10 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 22042 "TSS-I" 260.0 "COD-I" 657.0 "TKN" 65.5 "TSS-P" 130.0 "COD-P" 678.0 "MLSS-AT" 4160.0 "SVI-AT" 117.0 "SRT-AT" NIL "F-M-AT" NIL "Filam-Dominant-AT" NIL "DO-AT-1" 1.0 "DO-AT-2" 1.13 "TSS-E" 21.0 "COD-E" 76.0 "TN-E" 37.1 "Diagnosis" "bulking-sludge-filamentous" "WAS" 443.5 "RAS" 75.6 "Air-Flow-1" 105000 "Air-Flow-2" 87028)

↪ Identified by CBRS.

- ("Day" 8 "Month" 8 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 18575 "TSS-I" 90.0 "COD-I" 437.0 "TKN" NIL "TSS-P" 78.0 "COD-P" 420.0 "MLSS-AT" 2850.0 "SVI-AT" 175.4 "SRT-AT" 0.0 "F-M-AT" 0.0 "Filam-Dominant-AT" NIL "DO-AT-1" 2.82 "DO-AT-2" 2.5 "TSS-E" 41.0 "COD-E" 100.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 923.4 "RAS" 84.9 "Air-Flow-1" 75580 "Air-Flow-2" 70281)

↪ Identified by RBES.

- ("Day" 23 "Month" 8 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 17799 "TSS-I" 232.0 "COD-I" 751.0 "TKN" 95.8 "TSS-P" 98.0 "COD-P" 406.0 "MLSS-AT" 2470.0 "SVI-AT" 271.3 "SRT-AT" 8.16 "F-M-AT" 0.26 "Filam-Dominant-AT" NIL "DO-AT-1" 2.01 "DO-AT-2" 2.49 "TSS-E" 26.0 "COD-E" 86.0 "TN-E" 48.5 "Diagnosis" "bulking-sludge-filamentous" "WAS" 960.6 "RAS" 83.3 "Air-Flow-1" 161800 "Air-Flow-2" 127400)

↪ Identified by RBES.

G-2 set results.

- ("Day" 24 "Month" 11 "Year" 1999 "Hour" "10-0am" "Water-Flow-Rate-I" 21337 "TSS-I" 250.0 "COD-I" 791.0 "TKN" 133.0 "TSS-P" 78.0 "COD-P" 659.0 "MLSS-AT" 3248.0 "SVI-AT" 135.5 "SRT-AT" 8.0 "F-M-AT" 0.24 "Filam-Dominant-AT" NIL "DO-AT-1" 2.4 "DO-AT-2" 2.53 "TSS-E" 10.0 "COD-E" 77.0 "TN-E" 86.4 "Diagnosis" "bulking-sludge-filamentous" "WAS" 783.9 "RAS" 43.2 "Air-Flow-1" 0 "Air-Flow-2" 0)

  ↪ Identified by RBES.

- ("Day" 7 "Month" 5 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 22786 "TSS-I" 380.0 "COD-I" 878.0 "TKN" NIL "TSS-P" 152.0 "COD-P" 866.0 "MLSS-AT" 4330.0 "SVI-AT" 133.0 "SRT-AT" 6.42 "F-M-AT" 0.18 "Filam-Dominant-AT" NIL "DO-AT-1" 1.42 "DO-AT-2" 1.41 "TSS-E" 15.0 "COD-E" 86.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 613.6 "RAS" 76.2 "Air-Flow-1" 94313 "Air-Flow-2" 108600)

  ↪ Identified by RBES.

- ("Day" 26 "Month" 9 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 22755 "TSS-I" 300.0 "COD-I" 743.0 "TKN" NIL "TSS-P" 110.0 "COD-P" 570.0 "MLSS-AT" 3240.0 "SVI-AT" 184.7 "SRT-AT" 4.99 "F-M-AT" 0.0 "Filam-Dominant-AT" NIL "DO-AT-1" 2.0 "DO-AT-2" 2.01 "TSS-E" 49.0 "COD-E" 115.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 1338.7 "RAS" 67.6 "Air-Flow-1" 152000 "Air-Flow-2" 158800)

  ↪ Identified by CBRS and RBES.

- ("Day" 12 "Month" 9 "Year" 1999 "Hour" "10-0am" "Water-Flow-Rate-I" 16776 "TSS-I" 234.0 "COD-I" 650.0 "TKN" NIL "TSS-P" 100.0 "COD-P" 524.0 "MLSS-AT" NIL "SVI-AT" NIL "SRT-AT" NIL "F-M-AT" NIL "Filam-Dominant-AT" "Gordona" "DO-AT-1" 3.0 "DO-AT-2" 0.0 "TSS-E" 14.0 "COD-E" 77.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" NIL "RAS" NIL "Air-Flow-1" NIL "Air-Flow-2" NIL)

  ↪ Impasse situation.

- ("Day" 18 "Month" 8 "Year" 1999 "Hour" "10-0am" "Water-Flow-Rate-I" 16796.0 "TSS-I" 244.0 "COD-I" 671.0 "TKN"

NIL "TSS-P" NIL "COD-P" 542.0 "MLSS-AT" 3460.3 "SVI-AT" 123.9 "SRT-AT" NIL "F-M-AT" 0.15 "Filam-Dominant-AT" NIL "DO-AT-1" NIL "DO-AT-2" NIL "TSS-E" 16.0 "COD-E" 112.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" NIL "RAS" NIL "Air-Flow-1" NIL "Air-Flow-2" NIL)

↪ Impasse situation.

- ("Day" 30 "Month" 6 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 23055 "TSS-I" 444.0 "COD-I" 838.0 "TKN" NIL "TSS-P" 146.0 "COD-P" NIL "MLSS-AT" NIL "SVI-AT" NIL "SRT-AT" NIL "F-M-AT" NIL "Filam-Dominant-AT" NIL "DO-AT-1" 0.0 "DO-AT-2" 0.0 "TSS-E" 17.0 "COD-E" 73.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 676.3 "RAS" 100.0 "Air-Flow-1" 111000 "Air-Flow-2" 110000)

  ↪ Impasse situation.

- ("Day" 25 "Month" 8 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 18561 "TSS-I" 248.0 "COD-I" 718.0 "TKN" NIL "TSS-P" 148.0 "COD-P" NIL "MLSS-AT" NIL "SVI-AT" NIL "SRT-AT" 0.0 "F-M-AT" 0.0 "Filam-Dominant-AT" "Microthrix" "DO-AT-1" 2.02 "DO-AT-2" 2.02 "TSS-E" 34.0 "COD-E" 115.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 705.7 "RAS" 45.1 "Air-Flow-1" 76630 "Air-Flow-2" 77114)

  ↪ Impasse situation.

- ("Day" 7 "Month" 4 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 22247 "TSS-I" 364.0 "COD-I" 989.0 "TKN" 88.6 "TSS-P" 126.0 "COD-P" 664.0 "MLSS-AT" 4080.0 "SVI-AT" 130.7 "SRT-AT" 7.53 "F-M-AT" NIL "Filam-Dominant-AT" NIL "DO-AT-1" 1.15 "DO-AT-2" 1.15 "TSS-E" 16.0 "COD-E" 96.0 "TN-E" 47.6 "Diagnosis" "bulking-sludge-filamentous" "WAS" 661.5 "RAS" 100.3 "Air-Flow-1" 104500 "Air-Flow-2" 107200)

  ↪ Identified by RBES.

- ("Day" 10 "Month" 9 "Year" 1999 "Hour" "10-0am" "Water-Flow-Rate-I" 20638 "TSS-I" 278.0 "COD-I" 770.0 "TKN" NIL "TSS-P" 75.0 "COD-P" 625.0 "MLSS-AT" 3169.7 "SVI-AT" 134.1 "SRT-AT" NIL "F-M-AT" 0.54 "Filam-Dominant-AT" NIL "DO-AT-1" 3.0 "DO-AT-2" 0.0 "TSS-E" 14.0 "COD-E" 100.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" NIL "RAS" NIL "Air-Flow-1" NIL "Air-Flow-2" NIL)

  ↪ Identified by RBES.

- ("Day" 25 "Month" 5 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 26093 "TSS-I" 220.0 "COD-I" 636.0 "TKN" NIL "TSS-P" 66.0 "COD-P" 518.0 "MLSS-AT" 3320.0 "SVI-AT" 117.5 "SRT-AT" 6.58 "F-M-AT" 0.22 "Filam-Dominant-AT" NIL "DO-AT-1" 2.0 "DO-AT-2" 2.0 "TSS-E" 16.0 "COD-E" 92.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 758.9 "RAS" 100.2 "Air-Flow-1" 107000 "Air-Flow-2" 99904)

  ↪ Identified by CBRS.

G-3 set results.

- ("Day" 18 "Month" 7 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 20889 "TSS-I" 408.0 "COD-I" 990.0 "TKN" NIL "TSS-P" 152.0 "COD-P" 706.0 "MLSS-AT" 3990.0 "SVI-AT" 223.5 "SRT-AT" NIL "F-M-AT" NIL "Filam-Dominant-AT" NIL "DO-AT-1" 1.0 "DO-AT-2" 1.0 "TSS-E" 19.0 "COD-E" 66.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 564.2 "RAS" 106.0 "Air-Flow-1" 100400 "Air-Flow-2" 105700)

  ↪ Identified by CBRS and RBES.

- ("Day" 27 "Month" 7 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 21804 "TSS-I" 528.0 "COD-I" 696.0 "TKN" NIL "TSS-P" 146.0 "COD-P" 630.0 "MLSS-AT" 3530.0 "SVI-AT" 194.1 "SRT-AT" NIL "F-M-AT" NIL "Filam-Dominant-AT" NIL "DO-AT-1" 2.02 "DO-AT-2" 2.01 "TSS-E" 18.0 "COD-E" 93.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 728.8 "RAS" 70.6 "Air-Flow-1" 61568 "Air-Flow-2" 66264)

  ↪ Identified by CBRS and RBES.

- ("Day" 26 "Month" 7 "Year" 1999 "Hour" "10-0am" "Water-Flow-Rate-I" 18987.0 "TSS-I" 294.0 "COD-I" 867.0 "TKN" NIL "TSS-P" 109.0 "COD-P" 590.0 "MLSS-AT" 3298.5 "SVI-AT" 97.6 "SRT-AT" 19.59 "F-M-AT" 0.35 "Filam-Dominant-AT" NIL "DO-AT-1" NIL "DO-AT-2" NIL "TSS-E" 26.0 "COD-E" 102.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" NIL "RAS" NIL "Air-Flow-1" NIL "Air-Flow-2" NIL)

  ↪ Impasse situation.

- ("Day" 11 "Month" 8 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 18071 "TSS-I" 208.0 "COD-I" 554.0 "TKN" NIL "TSS-P" 106.0 "COD-P" NIL "MLSS-AT" NIL "SVI-AT" NIL

"SRT-AT" 0.0 "F-M-AT" 0.0 "Filam-Dominant-AT" NIL "DO-AT-1" 2.46 "DO-AT-2" 1.75 "TSS-E" 31.0 "COD-E" 92.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 1818.1 "RAS" 81.8 "Air-Flow-1" 149000 "Air-Flow-2" 138000)

↪ Impasse situation.

- ("Day" 29 "Month" 10 "Year" 1999 "Hour" "10-0am" "Water-Flow-Rate-I" 22298 "TSS-I" NIL "COD-I" NIL "TKN" NIL "TSS-P" 72.0 "COD-P" 541.0 "MLSS-AT" 4099.0 "SVI-AT" 133.7 "SRT-AT" NIL "F-M-AT" NIL "Filam-Dominant-AT" "Microthrix" "DO-AT-1" 3.55 "DO-AT-2" 4.85 "TSS-E" NIL "COD-E" NIL "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 1136.0 "RAS" 56.0 "Air-Flow-1" NIL "Air-Flow-2" NIL)

  ↪ Identified by RBES.

- ("Day" 16 "Month" 7 "Year" 1999 "Hour" "10-0am" "Water-Flow-Rate-I" 20652.0 "TSS-I" 248.0 "COD-I" 733.0 "TKN" NIL "TSS-P" 74.0 "COD-P" 521.0 "MLSS-AT" 2909.8 "SVI-AT" 138.9 "SRT-AT" 14.36 "F-M-AT" 0.33 "Filam-Dominant-AT" NIL "DO-AT-1" 1.6 "DO-AT-2" 2.21 "TSS-E" 15.0 "COD-E" 93.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 505.0 "RAS" 159.0 "Air-Flow-1" NIL "Air-Flow-2" NIL)

  ↪ Identified by RBES.

- ("Day" 24 "Month" 8 "Year" 1999 "Hour" "10-0am" "Water-Flow-Rate-I" 18284.0 "TSS-I" 414.0 "COD-I" 904.0 "TKN" NIL "TSS-P" 116.0 "COD-P" 543.0 "MLSS-AT" 1279.3 "SVI-AT" 161.2 "SRT-AT" NIL "F-M-AT" 0.71 "Filam-Dominant-AT" NIL "DO-AT-1" 3.0 "DO-AT-2" 3.0 "TSS-E" 7.0 "COD-E" 97.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" NIL "RAS" NIL "Air-Flow-1" NIL "Air-Flow-2" NIL)

  ↪ Identified by CBRS and RBES.

- ("Day" 11 "Month" 8 "Year" 1999 "Hour" "10-0am" "Water-Flow-Rate-I" 21140 "TSS-I" 298.0 "COD-I" 795.0 "TKN" 123.0 "TSS-P" 172.0 "COD-P" 723.0 "MLSS-AT" 2867.4 "SVI-AT" 156.5 "SRT-AT" 7.0 "F-M-AT" 0.4 "Filam-Dominant-AT" "Microthrix" "DO-AT-1" 2.11 "DO-AT-2" 2.12 "TSS-E" 15.0 "COD-E" 84.0 "TN-E" 74.0 "Diagnosis" "bulking-sludge-filamentous" "WAS" 798.0 "RAS" 103.0 "Air-Flow-1" NIL "Air-Flow-2" NIL)

  ↪ Identified by RBES.

- `("Day" 21 "Month" 5 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 17714 "TSS-I" 168.0 "COD-I" 395.0 "TKN" NIL "TSS-P" 70.0 "COD-P" NIL "MLSS-AT" 3550.0 "SVI-AT" 138.0 "SRT-AT" 6.58 "F-M-AT" 0.0 "Filam-Dominant-AT" NIL "DO-AT-1" 2.01 "DO-AT-2" 2.02 "TSS-E" 17.0 "COD-E" 82.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 396.6 "RAS" 101.5 "Air-Flow-1" 27193 "Air-Flow-2" 25231)`

  ↪ Identified by RBES.

- `("Day" 7 "Month" 9 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 22172 "TSS-I" 196.0 "COD-I" 641.0 "TKN" NIL "TSS-P" NIL "COD-P" NIL "MLSS-AT" NIL "SVI-AT" NIL "SRT-AT" NIL "F-M-AT" NIL "Filam-Dominant-AT" NIL "DO-AT-1" 2.53 "DO-AT-2" 2.0 "TSS-E" 32.0 "COD-E" 90.0 "TN-E" NIL "Diagnosis" "bulking-sludge-filamentous" "WAS" 741.1 "RAS" 79.4 "Air-Flow-1" 76295 "Air-Flow-2" 81904)`

  ↪ Impasse situation.

- `("Day" 20 "Month" 7 "Year" 2000 "Hour" "10-0am" "Water-Flow-Rate-I" 21059 "TSS-I" 344.0 "COD-I" 1080.0 "TKN" 83.7 "TSS-P" 182.0 "COD-P" 522.0 "MLSS-AT" 4080.0 "SVI-AT" 172.8 "SRT-AT" NIL "F-M-AT" NIL "Filam-Dominant-AT" NIL "DO-AT-1" 1.32 "DO-AT-2" 1.34 "TSS-E" 19.0 "COD-E" 86.0 "TN-E" 35.0 "Diagnosis" "bulking-sludge-filamentous" "WAS" 883.9 "RAS" 83.1 "Air-Flow-1" 110000 "Air-Flow-2" 96692)`

  ↪ Identified by CBRS and RBES.

Impasse situations correspond to a set of 10 days. For each of them we describe the advance in diagnosis due to the WaWO ontology. In doing this we have to take into account at least two circumstances: first, WaWO activates when an impasse situation has been reached and for this reason it includes weaker axioms with respect to the RBES; second, WaWO has usually at its disposal additional information about micro-organisms that we did not use in this evaluation because the RBES and the CBRS are not able to deal with it.

The two basic descriptors which are used in bulking diagnosis are *SVI-AT* and *Filam-Dominant-AT*. In general, WaWO not only try to detect filamentous-bacteria excessive proliferation, but offers also a specific actuation strategy according to the identified bacteria. In case the bacteria causing *Bulking-Sludge-Filamentous* are not determined, the *SVI-AT* is brought into play and a non specific solution (e.g., adding chemicals to increase the weight of the sludge flocs or eliminating all filamentous bacteria) is offered to avoid the consequences of bulking sludge. In the following part of the section an account of WaWO analysis for the impasse situations is given.

- ("Day" 25 "Month" 10 "Year" 1999)

  Microthrix, the value of *Filam-Dominant-AT*, is a subclass of *Filamentous-Bacteria*.

  Bulking-00 relation connects *Filamentous-Bacteria* to *Bulking-Sludge-Filamentous*, which is in WaWO a subclass of *WWTP-Operational-State*, that is the category used for diagnosis expression. *Bulking-Sludge-Filamentous*.

  Bulking-02 to *Bulking-20* relations connect, according to the particular dominant micro-organism (*Microthrix*in this case), the *Bulking-Sludge-Filamentous* class to a specific *Actuation*.

- ("Day" 23 "Month" 5 "Year" 2000)

  Same as 25.10.1999.

- ("Day" 16 "Month" 5 "Year" 2000)

  In WaWO the *SVI-AT* threshold for the detection of a bulking situation is lower than in the RBES. In the latter this value is 130, that is already lower than the usual 140-200 range. In WaWO it is 100 and this is due, as noted above, to the fact that OntoWEDSS did not find any other diagnostic up to this point.

  SVI-AT is a subclass of *Sludge-Volumetric-Index*.

  Bulking-01 relation connects *Sludge-Volumetric-Index* to *Bulking-Sludge*

  Bulking-21 relation connects *Bulking-Sludge* to a non specific *Actuation*: the destruction of filaments via chlorine addition up to 20 mg/L. If this actuation is not available for any reason, another relation connects to a second non specific actuation: the increase of the characteristic weight of flocs via inorganic coagulants addition, such as lime or ferric salts.

- (”Day” 12 ”Month” 9 ”Year” 1999)

  Same as 25.10.1999, with *Gordona* instead of *Microthrix*.

- (”Day” 18 ”Month” 8 ”Year” 1999)

  Same as 16.5.2000.

- (”Day” 30 ”Month” 6 ”Year” 2000)

  No diagnosis recommendation.

- (”Day” 25 ”Month” 8 ”Year” 2000)

  Same as 25.10.1999.

- (”Day” 26 ”Month” 7 ”Year” 1999), (”Day” 11 ”Month” 8 ”Year” 2000) and (”Day” 7 ”Month” 9 ”Year” 2000)

  No diagnosis recommendation.

Table 8.4: Evaluation of CBRS plus RBES plus WaWO.

Experiment	**Number of data**	**Correct classification**
G-1	8	100%
G-2	10	90%
G-3	11	73%

Mean successful outcomes: 88%

Standard deviation: 14%

Semi-difference between maximum and minimum values: 14%

The final evaluation result for the whole OntoWEDSS is presented in Table 8.4, where successful diagnostics coming from either one of the three systems are considered. Comparing this table with the previous ones, it is possible to notice that using only one technique, with the same data base, brings in general lower-grade diagnosis.

8.6 Discussion

Here, we have shown experiments about the *Bulking-Sludge* situation. Other experiments have been carried out with other kinds of problematic situations, such as *Underloading* and *Foaming-Sludge*, and similar results have been obtained. Testing the system with real data helped to identify needs of modifications and improvement. Certainly, a larger database is needed, as well as a more complete and more standard labelling of historical cases. Another issue to consider is that WWTPs work in *normal-operation* state most of the time and what is needed for learning are experiences of bad functioning.

In the evaluation experiments, OntoWEDSS has been able to successfully identify 88% of *Bulking-Sludge* situations, suggesting associated action strategies. We did not take into account situations incorrectly detected which, given the priority accorded to the CBRS, would have degraded the results. Nonetheless, the evaluation phase helped us to identify flaws in the conceptualization of the domain (categories, values' ranges, relations and problem-solving strategies) within each module of OntoWEDSS and allowed several improvements in the implementation. The experiential knowledge obtained from the evaluation of the performance of the overall system and the assessment we received from wastewater-treatment experts have been very useful to redesign certain parts of OntoWEDSS, in particular the decision-support layer.

Differently from a classic control system, that cannot easily identify problematic situations in which a lot of qualitative-information analysis is required, OntoWEDSS brings the possibility to systematically consider the whole available information. Thus, future work will focus on the inclusion, after a necessary modelling phase, of even more qualitative knowledge and on the implementation of more relations among the various descriptors defined in the ontology.

Chapter 9

Conclusions

> En el centro de la Selva existió hace mucho una extravagante familia de plantas carnívoras que, con el paso del tiempo, llegaron a adquirir conciencia de su extraña costumbre (...).
> Sensibles a la crítica, poco a poco fueron cobrando repugnancia a la carne hasta que llegó el momento en que (...) se negaron a comerla, asqueadas a tal grado que su simple vista les producía náuseas.
> Entonces decidieron volverse vegetarianas.
> A partir de ese día se comen únicamente unas a otras y viven tranquilas, olvidadas de su infame pasado.
>
> *Augusto Monterroso*

The **goals** of this thesis were: (a) the development of new and solid bridges between AI research and environment research, (b) the advancement in EDSS development, and (c) the use of ontologies in decision support for the solution of complex problems.

To meet these goals, the following **research objectives** were established and achieved:

1. The improvement of the modelling of the information about wastewater treatment processes (see §6). This include *solving* part of the existing *terminological*

confusion in the domain.

2. The incorporation of the microbiological knowledge related to the treatment process into the reasoning process (see §6.8). This *microbiological component* is modelled by an ontology. With an ontology, it is possible to capture, understand, describe and reason on the knowledge about the whole physical, chemical and microbiological environment of a WWTP.

3. The creation of a system to supervise the processes taking place at a wastewater treatment plant (see chapter 5). This system is an *environmental decision support system with three layers: perception, diagnosis and decision support*. A *more reliable management* with respect to previous systems is possible by means of the input of new knowledge (including microbiological knowledge), of a novel *integration between KBSs and ontologies*, and of the introduction of planning capabilities. The integration happens mainly at the diagnosis level, where the results of rule-based and case-based reasoning systems are compared before passing a final decision to the decision support layer.

4. The resolution of existing reasoning-impasses (see §6.4). New *solutions, to impasse situations* in previous systems, are found using the new knowledge encoded in the ontology. In the case of impasse (no diagnostics), the decision support system turns to the ontology, demanding an off-line diagnosis mainly based on its microbiological deep knowledge. The hierarchical structure and the axioms of the ontology, allowing reasoning on different levels of abstraction, help to diagnose the situation in case of impasse of other KBSs.

5. The representation of cause-effect relations in the wastewater domain (see §5.4.3). The decision support system formalizes a certain degree of *cause-effect reasoning*, thanks to the implementation of a set of axioms that enable the ontology to automatically deduce the answer to questions about the wastewater

domain.

6. The possibility to share or reuse in other plants the acquired knowledge about the management of a treatment plant (see §6.9). The decision support system is *portable* towards similar domains and creates an opportunity of knowledge sharing among various wastewater treatment plants. The introduction of the ontology is the key factor to help in *reusing knowledge* and facilitate the portability of the system. The ontology can be downloaded in several implementation languages thanks to Ontolingua translators. The lexicon and semantics of the ontology are as much standard as possible, synonyms are shown in the documentation and there are no hidden assumptions.

The **specific tasks**, undertaken to meet the objectives, and the results obtained are:

I. A preliminary study of the descriptors used in the wastewater domain has been carried out. A simple hybrid learning system for wastewater-treatment-processes' supervision has been proposed (Comas *et al.* 1999). The system includes the participation of human experts interacting with two clustering techniques applied to measured descriptors. The conclusions obtained from the analysis of only quantitative data were compared with the conclusions achieved after the addition to the analysis of qualitative data and results of microscopic observation. The improvement in diagnosis was evident.

II. A study of microbiological knowledge in the wastewater domain and an identification of most common micro-organisms have been carried out, to understand what can be the influence of biological variability. A set of microbiological features (see *qualitative descriptors* in Table 5.1) have been selected to be used in this thesis.

III. An ontology for the real-world domain of wastewater treatment processes have been created (Ceccaroni *et al.* 2000a). This ontology, WaWO, follows in its first formulation the design principles of Ontolingua and has been created through the on-line Ontolingua Ontology Editor, taking into account all the available, compatible indications on methodology coming from the *ontologies community*. WaWO is the manifestation of a shared understanding of the wastewater domain that is agreed among experts in environmental and chemical engineering.

IV. An initial prototype DSS for wastewater management is developed (Ceccaroni 2000; Ceccaroni *et al.* 2000b). Its architecture includes a rule-based expert system, a case-based reasoner and an ontology. This DSS is able to model the information about wastewater treatment processes.

V. The introduction of the WaWO ontology and the integration of different reasoning modules imposed the adaptation of previous work by people of the KEML group at the LSI department of UPC (Barcelona, Spain) and of the LEQUIA group at the Chemical and Environmental Engineering Laboratory of the University of Girona (Spain) about WWTP management. This adaptation consisted in the revision of previously used AI reasoning techniques (rule-based reasoning and case-based reasoning), to be able to build an efficiently-working integrated system.

VI. Methods to solve diagnosis impasses and to represent transitions between states of the domain have been defined as axioms. With them, we represented two kinds of cause-effect relations: association of micro-organisms to the problematic situations that they cause, and association of the actual state of the plant to the actions that need to be performed in order to reach the normal state from that actual state. These relations can then be used for some kind of qualitative simulation, in order to predict future states of a WWTP in response to alternative action schemas.

VII. A review of the impact of AI techniques on the definition and development of EDSSs is carried out (Cortés *et al.* 2000). The review includes a selection of successful applications to a wide range of environmental problems.

VIII. An environmental decision-support system, based on WaWO, for the improvement of the management of wastewater treatment plants has been created as an evolution of DAI-DEPUR+. The new system, OntoWEDSS, presents a novel general architecture, characterized by the integration of a domain *ontology*, a rule-based expert system, and a case-based reasoner. A coordination of priority exists among these KBSs at the diagnosis level.

IX. A reactive, linear planner, WaRP, is designed and applied to the wastewater domain as a component of the OntoWEDSS environmental decision-support system (Ceccaroni and Robertson 2000). WaRP is a real-time system employed to represent and use experts' procedural knowledge for accomplishing pre-defined goals. A working prototype has been implemented.

9.1 Future work and open research lines

In collaboration with biologists and chemical engineers, the evolution of the ontology in the OntoWEDSS decision support system from a superficial knowledge to a deeper knowledge of the microbiological component of the wastewater treatment process will be completed. This should allow an even better management of wastewater treatment plants.

A process of adaptation, update and refinement of current AI reasoning modules (rule- and case-based reasoning) of OntoWEDSS needs to be brought to a conclusion for the complete integration of the ontology into the resultant environmental decision-support system.

Open research lines to be considered, based on the experience from the Ph.D. thesis are:

- to be able to reason with variations/transitions of descriptors' values;
- to be able to simulate and predict the evolution of a treatment plant's state;
- Integration of the ontology with some temporal reasoning. For this, a primary task which we will address is that of temporal projection in wastewater treatment. This induces the following set of requirements on the ontology:
  - Temporal projection requires the evaluation of the truth value of a proposition at some point in time in the future. We therefore need to define axioms that express how the truth of a proposition changes over time. In particular, we need to address the frame problem and express the properties and relations that change or do not change as the result of an activity.
  - We must redefine the notion of a state of the WWTP, that is, define what is true of the WWTP before and after performing different activities. This is necessary to express the causal relationship between the preconditions and the effects of an activity (dynamic cause-effect relations).
  - The time interval over which the state has a certain status is bounded by the times at which the appropriate actions that change status occur. This interval defines the duration of a state. This is essential for the construction of schedules.
  - We want a uniform hierarchical representation for activities (aggregation). Plans and processes are constructed by combining activities. We must precisely define how activities are combined to form new ones. The

representation of these combined activities should be the same as the representation of the sub-activities. Thus, aggregate activities (sets of activities or processes) can themselves be represented as activities.

  - The causal and temporal structure of states and sub-activities of an activity should be explicit in the representation of the activity.

- Construction of a sub-ontology for knowledge management: this sub-ontology will connect *impasse situations* to the agents (people, systems) with the knowledge to solve them.

Cerca del Bosque de Chapultepec vivió hace tiempo un hombre que se enriqueció y se hizo famoso criando Cuervos para los mejores parques zoológicos del país y del mundo y los cuales resultaron tan excelentes que a la vuelta de algunas generaciones y a fuerza de buena voluntad y perseverancia ya no intentaban sacar los ojos a su criador sino que por lo contrario se especializaron en sacárselos a los mirones que sin falta y dando muestras del peor gusto repetían delante de ellos la vulgaridad de que no había que criar Cuervos porque le sacaban a uno los ojos.

Augusto Monterroso

Appendix A

WaWO's features

A.1 Identifying features

Ontology

- Name: WasteWater Ontology (WaWO)
- Server site: Ontolingua Ontology Editor (http://www-ksl-svc.stanford.edu)
- Web page: http://www.lsi.upc.es/∼luigic/WaWO/WaWO.htm, FAQs not available
- NL description: Ontology for wastewater-treatment plant management.
- Built date: 2000

Developer

- Name: Luigi Ceccaroni
- Web page: http://www.lsi.upc.es/∼luigic
- E-mail: luigic@lsi.upc.es
- Contact name: Luigi Ceccaroni
- Telephone: +34 . 93 401 56 46
- Fax: +34 . 93 401 70 14

- Postal address: Universitat Politècnica de Catalunya, Departament de Llenguatges i Sistemes Informàtics, Campus Nord, Modul C6, Jordi Girona 3, 08034 Barcelona, Spain

Distributor

- Name: UPC
- Web page: http://www.upc.es/∼ia
- E-mail: ia@lsi.upc.es
- Contact name: Ulises Cortés
- Telephone: +34 . 93 401 70 16
- Fax: +34 . 93 401 70 14
- Postal address: Universitat Politècnica de Catalunya, Departament de Llenguatges i Sistemes Informàtics, Campus Nord, Modul C5, Jordi Girona 3, 08034 Barcelona, Spain

A.2 Descriptive features

General

- Type of Ontology: domain-specific application
- Subject: management of wastewater treatment plants
- Purposes: systems modeling and engineering, knowledge reuse
- Ontological commitment (the nature of reality): facts in the world hold or do not hold; a degree of truth will be introduced in the LISP implementation.
- List of higher-level concepts: Actuation, Biological-Living-Object, Body-Of-Water, Descriptor, Eukaryotic-Cell, Fungus, Health, Output-Datum, Plan, Protozoan, State-Wwtp, Treatment, Virus, Wastewater-General, Wastewater-Treatment-Plant

- Implementation status: implemented and published (Ceccaroni *et al.* 2000a) in Ontolingua Language, planned in Allegro Common LISP
- On-line and hard copy documentation: available

Scope

- Number of concepts representing classes: 109
- Number of concepts representing instances: 0
- Number of explicit axioms: 0
- Number of relations: 1
- Number of functions: 15
- Number of class concepts at first, second and third level: 15, 14, 18
- Number of class leaves: 85
- Average branching factor:
- Average depth:
- Highest depth level: 6

Design

- Building methodologies: Ontolingua guidelines
- Steps followed: study of the potential utility, specification of purposes, knowledge acquisition, conceptualization, choice of development environment, decision about ontologies reuse, implementation, documentation and maintenance
- Level of formality of the methodology: semi-formal
- Building approach: top-down
- Level of specification formality: formal

- Knowledge sources: domain books, experts
- Reliability of knowledge sources: high
- Knowledge acquisition techniques: formal and informal text analysis, interviews
- Formalism paradigms: frame-based formalism
- Integrated ontologies: WaWO is a single ontology
- Languages in which the ontology is available: Ontolingua; formal knowledge-representation languages supported by available translators: KIF 3.0, CLIPS, CLIPS sentential format, CML ATP, CML rule engine, EpiKit, IDL, KSL rule engine, Loom, ProLog syntax

Requirements

- Hardware and software support: minimum disk space, RAM, processor and operating system, being WaWO stored remotely; an Internet browser

Cost

- Price of use: 0 euros
- Maintenance cost: 0 euros
- Estimated price of required software: 0 euros
- Estimated price of required hardware: 1000 euros

Usage

- Number of applications: 1 using WaWO as a source of knowledge
- List of main applications: OntoWEDSS

A.3 Functional features

- Description of use tools: taxonomic browser, on-line editor, translators, remote access via Internet browser
- Documentation quality: good
- Training courses: not necessary
- On-line help: Ontolingua general help
- Operating instructions: availability of instruction to remotely access, manipulate, display and update knowledge
- Availability of modular use: yes
- Possibility of adding new knowledge: yes
- Availability of PSMs: no

Appendix B

Sample database

Table B.1: Database schema of Granollers WWTP with sample values.

Descriptor name	Value
Acineria - Acineria uncinata - ?	0
Acineta - Acineta spp. - ?	0
Aspidisca - Aspidisca cicada - ?	0
Bacillus - Bacillus spp. - ?	0
BOD_5-E - effluent biochemical-oxygen-demand - mgO_2/L	24
BOD_5-I - inflow biochemical-oxygen-demand - mgO_2/L	420
BOD_5-P - primary-effluent biochemical-oxygen-demand - mgO_2/L	280
BOD_5/N - BOD_5/N ratio - ?	3
BOD_5/P - BOD_5/P ratio - ?	30
BOD_5R-B - BOD_5 removal after biological treatment - %	91
BOD_5R-E - Total BOD_5 removal - %	94
BOD_5R-P - BOD_5 removal after primary treatment - %	33
Chilodonella - Chilodonella uncinata - ?	0
Ciliates-D - Ciliates diversity - ?	1.16
Ciliates-T - total Ciliates - ?	6733
Ciliates-spp - unidentified Ciliates - ?	0
COD-E - effluent chemical-oxygen-demand - mgO_2/L	126
COD-I - inflow chemical-oxygen-demand - mgO_2/L	744
COD-P - primary-effluent chemical-oxygen-demand - mgO_2/L	406

Descriptor name	Value
COD/BOD_5-E - effluent COD/BOD_5 ratio - ()	5.25
COD/BOD_5-I - inflow COD/BOD_5 ratio - ()	1.77
COD/BOD_5-P - primary-effluent COD/BOD_5 ratio - ()	1.45
CODR-B - COD removal after biological treatment - %	68
CODR-E - Total COD removal - %	83
CODR-P - COD removal after primary treatment - %	45
Colpidium - Colpidium spp. - ?	0
Cond-E - inflow conductivity - μS/cm	4320
Cond-I - inflow conductivity - μS/cm	4150
Cond-P - inflow conductivity - μS/cm	4880
Day - day - dd/mm/yy	10/10/98
Diagnosis - diagnosis - qual.	Foaming-Sludge
Discophrya - Discophrya spp. -	0
DO-AT - biological-reactor dissolved oxygen - ?	3
Epistylis - Epistylis spp. - ?	1373
Euplotes - Euplotes spp. - ?	0
Filam-Dominant-AT - dominant filamentous bacteria - qual.	Microthrix
Filam-T-AT - total filamentous - m/mL	60
Filam-TI - total filamentous - intersections/mL	170,000
Flagellates-L - $\leq$ 20 μm flagellates - ?	41830
Flagellates-S - $>$ 20 μm flagellates - ?	65
Floc-F - filament effect on floc - qual.	null
Floc-Morphology-i - floc morphology i - qual.	weak
Floc-Morphology-ii - floc morphology ii - qual.	irregular
Floc-Morphology-iii - floc morphology iii - qual.	very-disperse
Floc-MS - floc mean size - μm	83
Floc-QA - floc qualitative assessment - qual.	bad
FlowRate-RAS - flow of recycled-activated-sludge - m^3/d	14,00

Descriptor name	Value
FlowRate-%REC - % of recycled water-flow - %	57
FlowRate-WAS - flow of waste-activated-sludge - m^3/d	36,00
F-M-AT - food-to-micro-organisms ratio - ?	0.78
Gimnamebas-S - $\leq 50\ \mu m$ Gimnamebas - ?	0
Gimnamebas-L - $> 50\ \mu m$ Gimnamebas -?	0
Gordona-A - Gordona abundance - ?	3
Gordona-spp - Gordona spp. - ?	165200
Haliscomenobacter - Haliscomenobacter hydrosis - ?	1
HRT - Hydraulic retention time - ?	0.3
Litonotus - Litonotus spp. - ?	0
MES-E - effluent MES - mg/L	31
MES-I - inflow MES - mg/L	242
MES-P - primary-effluent MES - mg/L	92
Metacineta - Metacineta spp. - ?	0
Microthrix - Microthrix parvicella - ?	13
MLSS-AT - biological-reactor mixed-liquor suspended solids - ?	1178
MLSS-RAS - recycled-activated-sludge mixed-liquor suspended solids - ?	8406
MLVSS-AT - biological-reactor mixed-liquor volatile suspended solids - %	77
MLVSS-RAS - RAS mixed-liquor volatile suspended solids - ?	——
Nematodes - Nematodes - ?	0
NH_4^+-E - effluent ammonium - mgN/L	50
NH_4^+-I - inflow ammonium - mgN/L	57
NO_2^--E - effluent nitrites - mgN/L	3.4
NO_2^--I - inflow nitrites - mgN/L	0.3
NO_3^--E - effluent nitrates - mgN/L	0.3
NO_3^--I - inflow nitrates - mgN/L	0.3
Nostocoida-I - Nostocoida limicola I - ?	0
Nostocoida-II - Nostocoida limicola II - ?	0

Descriptor name	Value
Nostocoida-III - Nostocoida limicola III - ?	0
ON-E - effluent organic-nitrogen - mgN/L	56
ON-I - inflow organic-nitrogen - mgN/L	110
ON-P - primary-effluent organic-nitrogen - mgN/L	——
Opercularia-spp - Other kinds of Opercularia spp. - ?	4706
Opercularia-a - Opercularia asymmetrica - ?	0
Parameci - Parameci spp. - ?	0
pH-E - effluent pH - ()	7.22
pH-I - inflow pH - ()	8.67
pH-P - primary-effluent pH - ()	8.51
PO_4^{3-}-E - effluent phosphates - mgP/L	——
PO_4^{3-}-I - inflow phosphates - mgP/L	——
Podophya - Podophya spp. - ?	0
Rotifers - Rotifers - ?	0
Sphaerotilus - Sphaerotilus natans - ?	45.6
SRT-AT - sludge residence time - ?	8
SVI-AT - sludge volumetric index - ?	169
Tecamebas - Tecamebas (Arcella) - ?	0
Teletrocs - Teletrocs - ?	0
Thiothrix-I - Thiothrix I - ?	0
Thiothrix-II - Thiothrix II - ?	0
Tipus-0041 - Tipus 0041 - ?	0
Tipus-0092 - Tipus 0092 - ?	0
Tipus-021N - Tipus 021N - ?	0
Tipus-0411 - Tipus 0411 - ?	0
Tipus-0581 - Tipus 0581 - ?	0
Tipus-0675 - Tipus 0675 - ?	0
Tipus-0803 - Tipus 0803 - ?	0

Descriptor name	Value
Tipus-0914 - Tipus 0914 - ?	0
Tipus-0961 - Tipus 0961 - ?	0
Tipus-1701 - Tipus 1701 - ?	0
Tipus-1863 - Tipus 1863 - ?	0
TKN - total Kjeldahl nitrogen - mgN/L	0.9
TN-E - effluent total-nitrogen - mgN/L	59
TN-I - inflow total-nitrogen - mgN/L	110
TP-E - effluent total-phosphor - mgP-$PO_4{}^{3-}$/L	3
TP-I - inflow total-phosphor - mgP-$PO_4{}^{3-}$/L	13
Trokophrya - Trokophrya - ?	0
TSS-E - effluent total-suspended-solids - mg/L	18
TSS-I - inflow total-suspended-solids - mg/L	300
TSS-P - primary-effluent total-suspended-solids - mg/L	110
TSSR-B - TSS removal after biological treatment - %	66
TSSR-E - Total TSS removal - %	87
TSSR-P - TSS removal after primary treatment - %	61
Turb-E - effluent turbidity - NTU	——
Turb-I - inflow turbidity - NTU	——
Turb-P - primary-effluent turbidity - NTU	——
Uronema - Uronema nigricans - ?	0
V30-RAS - RAS-activated-sludge volume settled in 30 min - ?	800
V30-AT - AT-activated-sludge volume settled in 30 min - ?	200
Vorticella-c - Vorticella convalaria - ?	0
Vorticella-i - Vorticella infosionum - ?	0
Vorticella-m - Vorticella microstoma - ?	654
Vorticella-s - Vorticella similis - ?	0
Vorticella-spp - Other kinds of Vorticella spp. - ?	0
Water-Flow-Rate-By-Pass - by-pass-to-river water-flow-rate - m^3/d	2

Descriptor name	Value
Water-Flow-Rate-E - effluent water-flow-rate - m^3/d	24,300
Water-Flow-Rate-I - inflow water-flow-rate - m^3/d	25,000
Water-Flow-Rate-P - primary-effluent water-flow-rate - m^3/d	24,500
Zn-I - inflow zinc - mgZn/L	———

Un día que el Zorro estaba muy aburrido y hasta cierto punto melancólico y sin dinero decidió convertirse en escritor, cosa a la cual se dedicó inmediatamente, pues odiaba ese tipo de personas que dicen voy a hacer esto o lo otro y nunca lo hacen.
Su primer libro resultó muy bueno, un éxito; todo el mundo lo aplaudió, y pronto fue traducido (a veces no muy bien) a los más diversos idiomas.
El segundo fue todavía mejor que el primero, y varios profesores norteamericanos (...) lo comentaron con entusiasmo y aun escribieron libros sobre los libros que hablaban de los libros del Zorro.
Desde ese momento el Zorro se dio con razón por satisfecho, y pasaron los años y no publicaba otra cosa.
Pero los demás empezaron a murmurar y a repetir "¿Qué pasa con el Zorro?", y cuando lo encontraban en los cócteles puntualmente se le acercaban a decirle tiene usted que publicar más.
-Pero si ya he publicado dos libros -respondía él con cansancio.
-Y muy buenos -le contestaban-; por eso mismo tiene usted que publicar otro.
El Zorro no lo decía, pero pensaba: "En realidad lo que éstos quieren es que yo publique un libro malo; pero como soy el Zorro, no lo voy a hacer.
Y no lo hizo.

Augusto Monterroso

References

Aamodt, A. (1989). Towards robust expert systems that learn from experience. an architectural framework. In *Proceedings of the 3rd European Knowledge Acquisition for Knowledge-Based Systems Workshop (EKAW-89)*, Paris, France, pp. 311–326.

Agre, G. (1996). A case-based integrated learning architecture. In D. Dochev, V. Stefanuk, and H. Ueno (Eds.), *Proceedings of the JCKBSE'96 - Second Joint Conference on Knowledge-Based Software Engineering*, Sozopol, Bulgaria, pp. 48–54.

Alamán, X., S. Romero, C. Aguirre, P. Serrahima, R. Muñoz, V. López, J. Dorronsoro, and E. de Pablo (1992). Mip: A real time expert system. In *Proceedings of the 8th Conf. On Artificial Intelligence Application (CAIA-92)*, Monterrey, CA, USA, pp. 302–302.

Allen, J., J. Hendler, and A. Tate (Eds.) (1990). *Readings in Planning.* Morgan Kaufmann Publisher.

Allen, J. F., R. Fikes, and E. Sandewall (Eds.) (1991). *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91).* Morgan Kaufmann.

Arpírez Vega, J. C., A. Gómez-Pérez, A. Lozano-Tello, and H. S. Pinto (2000). Reference Ontology and $(\text{ONTO})^2$Agent: The ontology yellow pages. *Knowledge And Information Systems (KAIS)*, **2**(4), 387–412.

Avouris, N. M. (1995). Co-operating knowledge-based systems for environmental decision-support. *Knowledge-Based Systems*, **8**(1), 39–53.

Baeza, J., E. Ferreira, and J. Lafuente (2000). Knowledge-based supervision and control of wastewater treatment plant: a real-time implementation. *Water Science & Technology*, **41**(12), 129–137.

Baker, P. G., C. A. Goble, S. Bechhofer, N. W. Paton, R. Stevens, and A. Brass (1999). An ontology for bioinformatics applications. *Bioinformatics*, **15**(6), 510–520.

Ball, J. A. (1984). Memes as Replicators. *Ethology and Sociobiology*, **5**, 145–61.

Barnett, M. W. and J. F. Andrews (1990). Knowledge based systems for operation of wastewater treatment processes. In *Proceedings of the 5th IAWPRC workshop on Instrumentation, Control and Automation of Water and Wastewater Treatment and Transport Systems*, Yokohama and Kyoto, Japan, pp. 211–218.

Bateman, J., B. Magnini, and G. Fabris (1995). *The Generalized Upper Model (GUM) knowledge-base: Organization and use*, pp. 60–72. In Mars (1995).

Beccari, M. (1991). Principi dei trattamenti biologici. In R. Canziani (Ed.), *Trattamento delle acque di rifiuto*, Milano, Italy, pp. 99–113.

Becker, B. (1987). The limits of knowledge acquisition. Technical Report FB-14, Dortmund University, Germany.

Benjamins, V. R., D. Fensel, S. Decker, and A. G. Pérez (1999). $(KA)^2$: Building ontologies for the internet: a mid term report. *International Journal of Human-Computer Studies*, **51**, 687–712.

Benjes, Jr., H. H. (1980). *Handbook of Biological Wastewater Treatment*. Water management series. Garland.

Borst, P. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Ph. D. thesis, University of Twente, The Netherlands.

Borst, P., H. Akkermans, and J. Top (1997). Engineering ontologies. *International Journal of Human-Computer Studies*, **46**(2-3), 365–406.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE J. Rob. Aut. (RA)*, **2**(1), 14–23.

Buchanan, B. and R. Smith (1988). Fundamentals of expert systems. *Annual Reviews on Computer Science*, **3**, 23–58.

Calori, G., F. Colombo, and G. Finzi (1994). Frame: a knowledge-base tool to support the choice of the right air pollution model. In G. Guariso and B. Page (Eds.), *Computer Support for Environmental Impact Assessment, Proceedings of the IFIP TC5/WG5.11 Working Conference on Computer Support for Environmental Impact Assessment (CSEIA 93)*, pp. 211–222.

Canziani, R. (Ed.) (1991). *Trattamento delle acque di rifiuto*. Istituto per l'Ambiente.

Ceccaroni, L. (2000). Integration of a rule-based expert system, a case-based reasoner and an ontological knowledge-base in the wastewater domain. In *Proceedings of ECAI 2000 - W07: Binding Environmental Sciences and Artificial Intelligence (BESAI2000)*, Berlin, Germany, pp. 8.1–8.10.

Ceccaroni, L., U. Cortés, and M. Sànchez-Marrè (2000b). Integrated management of wastewater treatment. In *Proceedings of the Congrés Català d'Intel·ligència Artificial (CCIA2000)*, Vilanova i la Geltrú, Spain, pp. 219–233.

Ceccaroni, L., U. Cortés, and M. Sànchez-Marrè (2000a). WaWO - An ontology embedded into an environmental decision-support system for wastewater treatment plant management. In *Proceedings of ECAI2000 - W09: Applications of ontologies and problem-solving methods*, Berlin, Germany, pp. 2.1–2.9.

Ceccaroni, L. and D. Robertson (2000). WaRP - A Reactive Planner integrated in an environmental decision-support system for Wastewater treatment plant management. In *Proceedings of ECAI 2000*, Berlin, Germany, pp. 491–495.

Chandrasekaran, B., R. Josephson, and V. R. Benjamins (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems*, **14**(1), 20–26.

Chaudhri, V. K., A. Farquhar, R. Fikes, P. Karp, and J. Rice (1998). OKBC: A programmatic foundation for knowledge base interoperability. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, Madison, WI, USA, pp. 600–607.

Chaudhri, V. K., A. Farquhar, R. Fikes, P. D. Karp, and J. P. Rice (1997). Open knowledge base connectivity 2.0. Technical Report KSL-98-06, Knowledge Systems Laboratory, Stanford, USA.

Christaller, T., F. Di Primio, and A. Vo' (Eds.) (1992). *The AI-Workbench BABYLON: An open and portable development environment for expert systems*. Academic Press.

Clancey, W. (1985). The epistemology of rule-based systems: a framework for explanation. *Artificial Intelligence*, **20**(3), 215–51.

Comas, J. (2000). *Development, implementation and evaluation of an activated sludge supervisory system for Granollers WWTP*. Ph. D. thesis, Departament d'enginyeria química agrària i tecnologies agroalimentàries, Universitat de Girona, Girona, Spain.

Comas, J., I. R-Roda, L. Ceccaroni, and M. Sànchez-Marrè (1999). Semi-automatic learning with quantitative and qualitative features. In *Proceedings of the VIII Conference of the Spanish Association for Artificial Intelligence (CAEPIA99)*, Murcia, Spain, pp. 17–25.

Cortés, U. and M. Sànchez-Marrè (Eds.) (1998). *Proceedings of the ECAI98 workshop: Binding environmental sciences and artificial intelligence (BESAI98)*, Brighton, UK.

Cortés, U., M. Sànchez-Marrè, L. Ceccaroni, I. R-Roda, and M. Poch (2000). Artificial intelligence and environmental decision support systems. *Applied Intelligence*, **13**, 77–91.

Cortés, U., M. Sànchez-Marrè, R. Sangüesa, J. Comas, I. R-Roda, M. Poch, and D. Riaño (2001). Knowledge management in environmental decision support systems. *AI Communications*, **14**, 3–12.

Cutkosky, M. R., R. S. Engelmore, R. E. Fikes, M. R. Genesereth, T. R. Gruber, W. S. Mark, J. M. Tenenbaum, and J. C. Weber (1993). PACT: An experiment in integrating concurrent engineering systems. *IEEE Computer*, **26**(1), 28–37.

Damiani, A. (1991). Caratteristiche dei liquami urbani. In R. Canziani (Ed.), *Trattamento delle acque di rifiuto*, Milano, Italy, pp. 11–42.

Dieng, R. *et al.* (Eds.) (2000). *Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW2000).* Lecture Notes in Artificial Intelligence (LNAI). Springer-Verlag.

Dym, C. L. and R. E. Levitt (1991). *Knowledge-based Systems in Engineering.* McGraw-Hill.

Efstathiou, J. and A. Mamdani (Eds.) (1985). *Expert Systems and Optimization in Process Control.* Technical Press.

Farquhar, A., R. Fikes, W. Pratt, and J. Rice (1995). Collaborative ontology construction for information integration. Technical Report KSL-95-63, Knowledge Systems Laboratory, Department of Computer Science, Stanford University, Stanford, USA.

Farquhar, A., R. Fikes, and J. Rice (1997). The ontolingua server: a tool for collaborative ontology construction. *Journal of Human-Computer Studies*, **46**, 707–728.

Fedra, K. (1993). *Expert systems in water resources simulation and optimization*, pp. 397–412. In Marco *et al.* (1993).

Fedra, K. (1995). Decision support for natural resources management: models, GIS and expert systems. *AI Applications*, **9**(3), 3–19.

Fensel, D. (2000). *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag.

Fensel, D., I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann, and M. Klein. *OIL in a nutshell*, pp. 1–16. In Dieng *et al.* (2000).

Fernández, M. (1999). Overview of methodologies for building ontologies. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR): Lessons Learned and Future Trends*, Stockholm, Sweden, pp. 4.1–4.13.

Fernández, M., A. Gómez-Pérez, and N. Juristo (1997). Methontology: from ontological art towards ontological engineering. In *Proceedings of AAAI-97 Spring*

Symposium on Ontological Engineering, Stanford, CA, USA, pp. 33–40.

Fikes, R. and A. Farquhar (1999). Distributed repositories of highly expressive reusable ontologies. *IEEE Intelligent Systems*, **14**(2), 73–79.

Fikes, R., A. Farquhar, and J. Rice (1997). Tools for assembling modular ontologies in ontolingua. Technical Report KSL-97-03, Knowledge Systems Laboratory, Department of Computer Science, Stanford University, Stanford, USA.

Finch, F. E., O. O. Oyeleye, and M. A. Kramern (1990). A robust event-oriented methodology for diagnosis of dynamic process systems. *Computers & Chemical Engineering*, **14**(12), 1379–1396.

Fine, S. S., C. J. Coats, Jr., A. Hanna, D. H. Loughlin, J. N. McHenry, R. Mathur, W. T. Smith, and N. J. M. Wheeler (1998). *The Environmental Decision Support System: A capable and versatile contribution to a community modeling and analysis system for air quality*, pp. 317–326. In Pepper *et al.* (1998).

Fridman-Noy, N. and C. D. Hafner (1997). The State of the Art in Ontology Design: A Survey and Comparative Review. *AI Magazine*, **18**(3), 53–74.

Gangemi, A., G. Steve, and F. Giacomelli (1996). Onions: An ontological methodology for taxonomic knowledge integration. In *Proceedings of the Workshop on Ontological Engineering, ECAI96*, Budapest, Hungary.

Genesereth, M. R. (1991). *Knowledge Interchange Format*, pp. 599–600. In Allen *et al.* (1991).

Giarratano, J. and G. Riley (1993). *Expert Systems Principles and Practice* (Second ed.). PWS Publishing. Second edition of *Expert Systems: Principles and Programming.*

Gómez-Pérez, A. (1994). Some ideas and examples to evaluate ontologies. Technical Report KSL-94-65, Knowledge Systems Laboratory, Department of Computer Science, Stanford University, Stanford, USA.

Gómez-Pérez, A. (1995). Criteria to verify knowledge sharing technology. Technical Report KSL-95-10, Knowledge Systems Laboratory, Department of Computer Science, Stanford University, Stanford, USA.

Gómez-Pérez, A. (1998). Knowledge sharing and reuse. *The Handbook of Applied Expert Systems*, 10.1–10.36.

Gómez-Pérez, A. and V. R. Benjamins (1999a). Applications of ontologies and problem-solving methods. *AI-Magazine*, **20**(1), 119–122.

Gómez-Pérez, A. and V. R. Benjamins (1999b). Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods(KRR5)*, Stockholm, Sweden, pp. 1.1–1.15.

Gómez-Pérez, A., M. Fernández, and A. de Vicente (1996). Towards a method to conceptualize domain ontologies. In *Proceedings of the workshop on Ontological Engineering, ECAI'96*, Budapest, Hungary, pp. 41–52.

González, A. J. and D. D. Dankel (1993). *The engineering of knowledge-based systems*. Prentice-Hall.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, **5**(2), 199–220.

Gruber, T. R. (1995). Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, **43**, 907–928.

Gruninger, M. and M. S. Fox (1994). The role of competency questions in enterprise engineering. In *Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice*, Trondheim, Norway, pp. 82–89.

Gruninger, M. and M. S. Fox (1995). Methodology for the design and evaluation of ontologies. In *Proceedings of the IJCAI1995 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada.

Guarino, N. (1998). Formal Ontology and Information Systems. In *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems (FOIS'98)*, Trento, Italy, pp. 3–15.

Guarino, N. and P. Giaretta (1995). *Ontologies and Knowledge Bases: Towards a Terminological Clarification*, pp. 25–32. In Mars (1995).

Guarino, N. and R. Poli (1995). The Role of Ontology in the Information Technology. *International Journal of Human-Computer Studies*, **43**(5-6), 623–965.

Guariso, G. and H. Werthner (Eds.) (1989). *Environmental Decision Support Systems*. Ellis Horwood - Wiley.

Guha, R. V. and D. B. Lenat (1990). Cyc: A Midterm Report. *AI Magazine*, **11**(3), 32–59.

Hartvigsen, G. and D. Johansen (1990). Cooperation in a distributed artificial intelligence environment - the StormCast application. *Eng. Appli. of AI*, **3**, 229–237.

Hayes-Roth, F. (1984). The knowledge-based expert system: a tutorial. *IEEE Computer*, **17**(9), 11–28.

Hushon, J. M. (1987). Expert systems for environmental problems. *Environmental Science & Technology*, **21**(9), 838–41.

Jackson, P. (1990). *Introduction to expert systems*. Addison Wesley.

Jenkins, D., M. G. Richard, and G. T. Daigger (1993). *Manual on the causes and control of activated sludge bulking and foaming* (Second ed.). Lewis Publishers.

Jones, D. M., T. J. M. Bench-Capon, and P. R. S. Visser (1998). Methodologies for Ontology Development. In *Proceedings of IT&KNOWS - Information Technology and Knowledge Systems, 15th IFIP World Computer Congress*, Vienna, Austria and Budapest, Bulgaria.

Joslyn, C. (1994). A possibilistic approach to qualitative model-based diagnosis. *Telematics and Informatics*, **11**(4), 365–384.

Jørgensen, S. E. and I. Johnsen (1989). *Principles of environmental science and technology* (Second ed.). Elsevier Science.

Kaelbling, L. P. *An architecture for intelligent reactive systems*, pp. 713–728. In Allen *et al.* (1990).

Kalfoglou, Y. (2000). *Deploying Ontologies in Software Design*. Ph. D. thesis, University of Edinburgh, Edinburgh, UK.

Karp, P., K. Myers, and T. Gruber (1995). The generic frame protocol. In *Proceedings of the 1995 International Joint Conference on Artificial Intelligence (IJCAI95)*, Montreal, Canada, pp. 768–774.

Kifer, M., G. Lausen, and J. Wu (1995). Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, **42**(4), 741–843.

Klein, M., D. Fensel, F. van Harmelen, and I. Horrocks (2000). The relation between ontologies and schema-languages: Translating OIL-specifications in XML-Schema. In *Proceedings of ECAI2000 - W09: Applications of ontologies and problem-solving methods*, Berlin, Germany, pp. 7.1–7.12.

Knight, K. and S. Luk (1994). Building a large knowledge base for machine translation. In *Proceedings of the American Association of Artificial Intelligence Conference - AAAI 94*, Seattle, USA, pp. 773–778.

Krovvidy, S. and W. G. Wee (1991). Wastewater treatment systems from case-based reasoning. *Machine Learning*, **10**, 341–363.

Lenat, D. and R. V. Guha (1990). *Building Large Knowledge Based Systems: Representation and Inference in the Cyc Project*. Addison Wesley.

Lester, J. (1994). *Generating Natural Language Explanations from large-scale Knowledge Bases*. Ph. D. thesis, Computer Science Department, Univ. Texas, Austin, TX, USA.

Loughlin, D. H. (1998). *Genetic Algorithm-Based Optimization in the Development of Tropospheric Ozone Control Strategies: Least Cost, Multiobjective, Alternative Generation, and Chance-Constrained Applications*. Ph. D. thesis, Civil Engineering Dept., North Carolina State University, Raleigh, NC, USA.

MacGregor, R. (1991). *The evolving technology of classification-based knowledge representation systems*, pp. 385–400. In Sowa (1991).

Madoni, P. (1994). A sludge biotic index (SBI) for the evaluation of the biological performance of activated sludge plants based on the microfauna analysis. *Wat. Res.*, **28**(1), 67–75.

Marco, J. B., R. Harboe, and J. Salas (Eds.) (1993). *Stochastic hydrology and its use in water resources systems simulation and optimization*. Kluwer Academic.

Marir, F. and I. Watson (1994). Case-based reasoning: A categorised bibliography. *The Knowledge Engineering Review*, **9**(3), 382–419.

Mars, N. J. I. (Ed.) (1995). *Towards Very Large Knowledge Bases. Proceedings of the 2nd International Conference on Knowledge Sharing (KB and KS'95)*, Twente, The Netherlands. IOS Press.

Marsili-Libelli, S. (1998). Adaptive fuzzy monitoring and fault detection. *Int. J. COMADEM*, **1**(3), 31–38.

Mason, C. (Ed.) (1995). *Proceedings of the IJCAI95 workshop on Artificial Intelligence and the Environment*, Montreal, Canada.

Matwin, S., D. Charlebois, D. G. Goodenough, and P. Bhogal (1995). Machine learning and planning for data management in forestry. *IEEE Expert Systems*, **10**(5), 35–40.

McGuire, J. G., R. N. Pelavin, J. C. Weber, J. M. Tenenbaum, T. R. Gruber, and G. R. Olsen (1993). SHADE: A Medium for Sharing Design Knowledge among Engineering Tools. *Concurrent Engineering: Applications and Research (CERA)*, **1**(3), 137–146.

Miller, G. A. (1990). Wordnet: an online lexical database. *International Journal of Lexicography*, **3**(4), 235–312.

Monod, J. (1942). *Recherches sur la croissance des cultures bactériennes*. Hermann and Cie.

Neches, R., R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swartout (1991). Enabling Technology for Knowledge Sharing. *AI Magazine*, **12**(3), 36–56.

Newell, A. (1990). *Unified theories of cognition*. Harvard University Press, Cambridge, MA, USA.

Novotny, V., H. Jones, X. Feng, and A. G. Capodaglio (1990). Time Series Analysis Models of Activated Sludge Plants. *Water Science & Technology*, **23**(4-6), 1107–1116.

Noy, N. F., R. W. Fergerson, and M. A. Musen (2000). The knowledge model of protege-2000: Combining interoperability and flexibility. In *Proceedings of the 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*, Juan-les-Pins, France.

O'Leary, D. (1997). Impediments in the use of explicit ontologies for kbs development. *International Journal of Human-Computer Studies*, **46**(2), 327–337.

Olsson, G. and B. Newell (1999). *Wastewater Treatment Systems. Modelling, Diagnosis and Control.* IWA Publishing.

Ortolano, L., G. LeCoeur, and R. MacGilchrist (1990). Expert System for Sewer Network Maintenance: Validation Issues. *Journal of Computing in Civil Engineering*, **4**(1), 37–54.

Patry, G. and D. Chapman (Eds.) (1989). *Dynamic modelling and expert systems in wastewater engineering*. Lewis Publisher.

Pepper, D. W., C. A. Brebbia, and P. Zannetti (Eds.) (1998). *Development and Application of Computer Techniques to Environmental Studies*. WIT Press.

Plaza, E., A. Aamodt, A. Ram, W. van der Velde, and M. van Someren (1993). Integrated learning architectures. In *Proceedings of the European Conference on Machine Learning (ECML1993)*, Volume 667 of *Lecture Notes in Artificial Intelligence (LNAI)*, pp. 429–441.

Puyol, J. and C. Sierra (1997). MILORD II: A Language Description. *Mathware and Soft Computing*, **4**, 299–338.

Quine, W. v. O. (1992). *Theories and Things* (Revised ed.). Harvard University Press.

Riaño, D. (1998). Learning rules within a framework of environmental sciences. In *ECAI 98 - W7 (BESAI98) workshop notes*, Brighton, UK, pp. 151–165.

Rickel, J. and B. Porter (1997). Automated Modeling of Complex Systems to Answer Prediction Questions. *Artificial Intelligence*, **93**, 201–260.

Robertson, D. (1998). Pitfalls of formality in early system design. In *Proceedings of the 1998 ARO/NSF Monterrey Workshop on Increasing the Practical Impact of Formal Methods for Computer-Aided Software Development*, Carmal, CA, USA.

Sanz, R., J. Aguilar, C. Sierra, L. Godó, and A. Ollero (1988). Adaptive control with a supervisor level using a rule-based inference system with approximate reasoning. In *Artificial Intelligence in Scientific Computation: towards 2nd generation systems, Proceedings of IMAC88*, Paris, France, pp. 337–342.

Sazonova, L. and G. Osipov (1998). Intelligent system for fish stock prediction and allowable catch evaluation. In *ECAI 98 - W7 (BESAI98) workshop notes*, Brighton, UK, pp. 181–96.

Schank, R. S. (1982). *Dynamic memory: a theory of learning in computers and people*. Cambridge University Press.

Schreiber, A. T., B. Wielinga, and W. H. J. Jansweijer (1995). The KACTUS view on the 'O' word. In *Proceedings of the IJCAI1995 Workshop on Basic Ontological Issues in Knowledge Sharing*. Montreal, Canada.

Schreiber, G., H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. van der Velde, and B. Wielinga (1999). *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press.

Serra, P. (1993). *Development of a knowledge-based system for control and supervision of urban wastewater treatment plants*. Ph. D. thesis, Chemistry Department, UAB, Barcelona, Spain.

Shepherd, A. and L. Ortolano (1996). Water-supply system operations: Critiquing expert-system approach. *J. Of Water Resources Planning and Management*, **122**(5), 348–355.

Shortliffe, E. H. and B. G. Buchanan (1975). A model of inexact reasoning in medicine. *Mathematical Biosciences*, **23**, 351–379.

Sànchez-Marrè, M. (1995). *DAI-DEPUR: an integrated supervisory multi-level architecture for wastewater treatment plants*. Ph. D. thesis, Software Department, UPC, Barcelona, Spain.

Sànchez-Marrè, M., U. Cortés, J. Lafuente, I. R-Roda, and M. Poch (1996). DAI-DEPUR: an integrated and distributed architecture for wastewater treatment plants supervision. *Artificial Intelligence in Engineering*, **10**, 275–285.

Sànchez-Marrè, M., U. Cortés, I. R-Roda, and M. Poch (1998). Case learning through a similarity measure for continuous domains. In *ECAI 98 - W7 (BESAI98) workshop notes*, Brighton, UK, pp. 39–53.

Sànchez-Marrè, M., U. Cortés, I. R-Roda, and M. Poch (1999). Sustainable case learning for continuous domains. *Environmental Modelling & Software*, **14**, 349–57.

Sànchez-Marrè, M., U. Cortés, I. R-Roda, M. Poch, and J. Lafuente (1997). Learning and adaptation in wastewater treatment plants through case-based reasoning. *Microcomputers in Civil Engineering*, **12**(4), 251–266.

Sowa, J. (Ed.) (1991). *Principles of semantic networks: explorations in the representation of knowledge*. Morgan Kaufmann.

Sowa, J. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole.

Sriram, D. and R. A. Adey (Eds.) (1986). *Applications of Artificial Intelligence in Engineering Problems*. Springer-Verlag.

Staab, S. and A. Maedche (2000). Ontology engineering beyond the modeling of concepts and relations. In *Proceedings of ECAI2000 - W09: Applications of ontologies and problem-solving methods*, Berlin, Germany, pp. 8.1–8.6.

Steele, Jr., G. L. (1990). *Common Lisp :The Language* (Second ed.). Digital Press.

Steels, L. (1990). Components of expertise. *AI Magazine*, **11**(2), 28–49.

Stefik, M., J. Aikins, R. Balzer, J. Benoit, F. Birnbaum L., Hayes-Roth, and E. Sacerdoti (1982). The organization of expert systems: a tutorial. *Artificial Intelligence*, **18**, 135–73.

Stephanopoulos, G. (1990). Artificial intelligence in process engineering: current state and future trends. *Computers & Chemical Engineering*, **14**, 1259–70.

Stuckenschmidt, H. (2000). Using OIL for Intelligent Information Integration. In *Proceedings of ECAI2000 - W09: Applications of ontologies and problem-solving methods*, Berlin, Germany, pp. 9.1–9.10.

Studer, R., V. R. Benjamins, and D. Fensel (1998). Knowledge engineering: principles and methods. *Data and Knowledge Engineering*, **25**(1-2), 161–197.

Swartout, B., R. Patil, K. Knight, and T. Russ (1996). Towards distributed use of large-scale ontologies. In *Proceedings of 10th Knowledge Acquisition Workshop (KAW1996)*, Banff, Canada, pp. 32.1–32.19.

Tandoi, V. (1991). *Trattamento a fanghi attivi - Gestione del processo e aspetti microbiologici*, pp. 165–184. In Canziani (1991).

Tomei, M. C. and A. C. Di Pinto (1991). *Il controllo del processo a fanghi attivi: stato dell'arte*, pp. 465–491. In Canziani (1991).

U.S. EPA (1977). *Wastewater treatment facilities for sewered small communities. Process design manual.* EPA. EPA/625/1-77/009.

Uschold, M. (1998a). Knowledge level modelling: concepts and terminology. *The Knowledge Engineering Review*, **13**(1), 5–29.

Uschold, M. (1998b). Where are the killer apps? In *Proceedings of the ECAI'98 Workshop on Applications of Ontologies and Problem Solving Methods*, Brighton, UK.

Uschold, M., P. Clark, M. Healy, K. Williamson, and S. Woods (1998). An Experiment in Ontology Reuse. In *Proceedings of 10th Knowledge Acquisition, Modeling and Management Workshop (KAW1998)*, Banff, Canada.

Uschold, M. and M. Gruninger (1996). Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, **11**(2), 93–136.

Uschold, M. and R. Jasper (1999). A framework for understanding and classifying ontology applications. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods(KRR5)*, Stockholm, Sweden, pp. 11.1–11.12.

Uschold, M., M. King, S. Moralee, and Y. Zorgios (1998). The enterprise ontology.

The Knowledge Engineering Review, **13**(1), 31–89. Also available as AIAI-TR-195 from AIAI, University of Edinburgh.

Uschold, M. and A. Tate (1998). Putting Ontologies to Use. *The Knowledge Engineering Review*, **13**(1), 1–3.

van Heijst, G., A. T. Schreiber, and B. J. Wielinga (1997). Using explicit ontologies in kbs development. *International Journal of Human-Computer Studies*, **46**(2-3), 183–292.

van Heijst, G., A. T. Schreider, and B. J. Wielinga (1997). Using explicit ontologies in kbs development. *International Journal of Human-Computer Studies*, **46**(2-3), 183–292.

van Lehn, K. (Ed.) (1990). *Architectures for intelligence*. Lawrence Erlbaum.

Venkatasubramanian, V. (1994). Towards integrated process supervision: current status and future directions. In *Proceedings of the 2nd IFAC workshop on Computer Software Structures Integrating AI / KBS Systems in Process Control*, pp. 9–21. Lund, Sweden.

Wanner, J. (1994). *Activated Sludge Bulking and Foaming Control*. Technomic Publishing.

Warren, D. H. D. and F. C. N. Pereira (1982). An efficient easily adaptable system for interpreting natural language queries. *Computational Linguistics*, **8**(3-4), 110–122.

Wen, C.-H. and C. A. Vassiliadis (1998). Applying hybrid artificial intelligence techniques in wastewater treatment. *Engineering Applications of Artificial Intelligence*, **11**, 685–705.

Wiener, N. (1961). *Cybernetics (2nd ed.)*. MIT Press.

Wright, J., L. Wiggins, R. Jain, and T. John Kim (Eds.) (1993). *Expert systems in Environmental Planning*. Springer-Verlag.

Yang, C. T. and J. J. Kao (1996). An expert-system for selecting and sequencing wastewater treatment processes. *Water Science & Technology*, **34**(3-4), 347–353.

Zhu, X. X. and A. R. Simpson (1996). Expert system for water treatment plant operation. *Journal of Environmental Engineering*, **122**(9), 822–829.