Reasoning About Actions for the Management of Urban Wastewater Systems: Preliminary report

Juan Carlos NIEVES [a,1], Montse AULINAS [a,b] and Ulises CORTÉS [a]

[a] *Knowledge Engineering and Machine Learning Group*
Technical University of Catalonia, Barcelona, Spain

[b] *Laboratory of Chemical and Environmental Engineering*
University of Girona, Spain

Abstract. It is well-known that the management of Urban Wastewater Systems (UWS) is a complex and critical process. To decide which is the correct sequence of actions for managing a given circumstance it is necessary a sophisticated analysis of hypothetical impact of these. Hence, the design of intelligent systems able to perform temporal projections based on the description of actions could help to deal with risk scenarios.

In this paper, we propose to use recent developments in knowledge representation languages and non-monotonic reasoning methodologies for representing and reasoning about actions for the Management of Urban Wastewater Systems. To this end, we explore the use of an action representation language called $\mathcal{A}$ for reasoning about the actions in the management of UWS. In particular, we consider the problem of industrial discharges. We present a declarative representation of a transition function from states and actions to states which allows us to make conclusions about a particular situation that may arise if one performs a particular sequence of actions.

Keywords. Wastewater Management, Reasoning About Actions

Introduction

Environmental decision-making is a complex, multidisciplinary and crucial task. As an example, in the water management field, water managers have to deal with complex problems due to the characteristics of processes that occur within environmental systems. Some of the common and special problematic features of environmental processes are [12]:

- Their intrinsic instability;
- Many of the facts and principles underlying the domain cannot be characterized precisely solely in terms of a mathematical theory or a deterministic model;

[1]Correspondence to: Knowledge Engineering and Machine Learning Group Technical University of Catalonia, Campus Nord - Edifici C5, Jordi Girona 1-3, 08034, Barcelona, Spain; E-mail: J.C. Nieves: jcnieves@lsi.upc.edu; M. Aulinas: aulinas@lequia.udg.cat; U. Cortés: ia@lsi.upc.edu

- Uncertainty and imprecision of data or approximate knowledge and vagueness: some of the environmental processes generate a considerable amount of qualitative information;
- Huge quantity of data/information; and,
- Heterogeneity and multiplicity of scales.

Specifically, one of the major problems that water managers have to face is water pollution. The management of polluted water in Urban Wastewater Systems (UWS) requires *planning* and *management activities* involving several sectors such as environment, energy, industries, *etc*. For centralized Wastewater Treatment Plants (WWTPs) for mixed household and industrial wastewater, managing the interactions between sources and treatment plant becomes a key issue for a successful operation of the treatment plant and to avoid deterioration of the river. Especially the contribution from industries must be regulated properly in order to avoid operational problems at the central treatment plant and transfer of toxic or persistent substances to end up in the effluent or sludge [9].

Assessment of industrial discharges to WWTPs is a complex issue where a lot of information and complex knowledge needs to be managed. For that reason it might be relevant to supply information on the priorities in a relatively simple way. In order to deal with these complex environmental data and knowledge above described, suitable knowledge representation tools are needed. They must allow to model complex biological systems and their behaviour (*i.e.* the biological processes at activated sludge WWTPs, or the expression of causal relationships at the UWS). Dworschak *et al.* in [4] reports the manifold advantages of action languages for modelling biological systems. As follows:

- The possibility to get a simple model.
- The possibility to use different kinds of reasoning to plan and support the experiments.
- The possibility to predict consequences and explanation of observations (by using further reasoning modes).
- The possibility to use static causal laws that allow to easily include background knowledge *e.g.* environmental conditions such as temperature, that play an important role for the development of biological systems such as the activated sludge processes at WWTP, difficult to include in the formal model.
- The possibility to easily extend a part of the model without requiring to change the rest of it.

Nowadays, action languages have received considerable attention in the knowledge representation and non-monotonic reasoning community. In fact their properties (complexity, expressiveness, *etc*.) have been studied in depth. Some of these action languages are $\mathcal{A}$ [7] and DLV^K [5]. In this paper, we explore $\mathcal{A}$ which is significantly different from the strict operator-based frameworks of STRIPS and PDDL [11]. In particular, we formalize a transition function under the $\mathcal{A}$'s syntax in order to formulate temporal projections of the impact of a sequence of actions for managing industrial discharges. Since our application domain is pervaded with incomplete information we extend the $\mathcal{A}$'s syntax in order to capture incomplete information. Essentially, we consider disjunctions in the syntax of effect propositions.

Given that $\mathcal{A}$'s inferences can be captured by the answer set semantics [6], we have implemented an Answer Set Program (ASP) which can compute our transformation function. This means that we can incorporate our transition function into real-systems by considering answer set solvers as SMODELS [13].

The rest of the paper is organized as follows: in §1, the basic syntax of $\mathcal{A}$ is presented. In §2, we introduce a finite state machine which defines the finite set of fluents and actions of a transition function from states and actions to states. In §3, we present the representation of the finite state machine introduced in §2 in terms of the $\mathcal{A}$'s syntax. Finally, in the last section, we outline our conclusions and future work.

1. The language $\mathcal{A}$

$\mathcal{A}$ was initially introduced by Gelfond and Lifschitz in [7]. However, it has been extended in several directions to incorporate additional features of dynamic worlds [1,4]. For instance, in [4] a variant of $\mathcal{A}$ is used for capturing biological Networks and in [2] a variant of $\mathcal{A}$ is used for capturing signalling networks. The following basic definitions were taken from [1].

The alphabet of the language $\mathcal{A}$ consists of two nonempty disjoint sets of symbols **F** and **A**. They are called the set of fluents **F** and the set of actions **A**. A *fluent* expresses the property of an object in a world, and forms part of the description of states of this world. A *fluent literal* is a fluent or a fluent preceded by $\neg$. A *state* σ is a collection of fluents. We say a fluent f holds in a state σ if $f \in \sigma$. We say a fluent literal $\neg f$ holds in σ if $f \notin \sigma$.

Situations are representations of the history of action execution. In the initial situation no action has been executed: we represent this by the empty list []. The situation $[a_n, \ldots, a_1]$ corresponds to the history where action a_1 is executed in the initial situation followed by a_2, and so on until a_n. There is a simple relation between situation and state. In each situation s certain fluents are true and certain others are false, and this is a *state of the world*.

Usually, the syntax of $\mathcal{A}$ is presented in three sub-languages:

Domain description: The domain description language is used to succinctly express the transition between states due to actions. It consists of effect propositions of the following form: **(1)** a ***cause*** $f_1 \vee \cdots \vee f_k$ ***if*** $p_1, \ldots, p_n, \neg q_{n+1}, \ldots, \neg q_r$
where a is an action, $f_i (1 \leq i \leq k)$ is a fluent literal, and $p_1, \ldots, p_n$ and $q_{n+1}, \ldots q_r$ are fluents. Intuitively, the above proposition means that if the fluent literals $p_1, \ldots, p_n, \neg q_{n+1}, \ldots, \neg q_r$ hold in the state corresponding to a situation s then in the state corresponding to a situation reached by executing a (denoted by $[a|s]$) a fluent literal f_i must hold. If both n and r are equal to 0 in (1) then we simply write: **(2)** a ***cause*** $f_1 \vee \cdots \vee f_k$
It is worth mentioning that the original syntaxis of the effect propositions is $k = 1$ [1,7]; however, we will see that the consideration of disjunctions is a straightforward way to capture incomplete information.

Observation language: A set of observations O consists of value propositions of the following form: **(3)** f ***after*** $a_1, \ldots, a_m$
where f is a fluent literal and $a_1, \ldots, a_m$ are actions. Intuitively, the above value proposition means that if $a_1, \ldots, a_m$ would be executed in the initial situation then in the state corresponding to the situation $[a_m, \ldots, a_1]$, f would hold. When $a_1, \ldots, a_m$ is an empty sequence, we write the above as follows:

(4) ***initially*** f

In this case the intuitive meaning is that f holds in the initial state corresponding to the initial situation.

Query language: Queries consists of value propositions of the form (3)

The role of effect propositions is to define a *transition function* from states and actions to states. Given a domain description D, such a transition function Φ should satisfy the following properties. For all actions a, fluents f and states σ:

- if D includes an effect proposition of the form (1) where f is the fluent g and $p_1, \ldots, p_n, \neg q_{n+1}, \ldots, \neg q_r$ hold in σ then $g \in \Phi(a, \sigma)$;
- if D includes an effect proposition of the form (1) where f is a negative fluent literal $\neg g$ and $p_1, \ldots, p_n, \neg q_{n+1}, \ldots, \neg q_r$ hold in σ then $g \notin \Phi(a, \sigma)$; and
- if D does not include such effect propositions, then $g \in \phi(a, \sigma)$ iff $g \in \sigma$.

If such a transition function exists, then we say that D is *consistent* and refer to its transition function by Φ_D. Given a consistent domain description D the set of observations O is used to determine the states corresponding to the initial situation, referred as the initial states and denoted by σ_0. While D determines a unique transition function, an O may not always lead to a unique initial state.

We say σ_0 is a initial state corresponding to a consistent domain description D and a set of observations O, if for all observations of the form (3) in O, the fluent literal f holds in the state $\Phi(a_m, \Phi(a_{m-1}, \ldots, \Phi(a_1, \sigma_0) \ldots))$. We will denote this state by $[a_m, \ldots, a_1]\sigma_0$. We say that (σ_0, Φ_D) *satisfies* O.

Given a consistent domain description D and a set of observation O, we refer to the pair (Φ_D, σ_0), where Φ_D is the transition function of D and σ_0 is an initial state corresponding to (D, O), as a model of (D, O). We say that (D, O) is *consistent* if it has a model and say it is *complete* if it is has a unique model.

We say a consistent domain D in the presence of a set of observations O entails a query Q of the form (3) if for all initial states σ corresponding to (D, O), the fluent literal f holds in the state $[a_m, \ldots, a_1]\sigma_0$. We denote this as $D \models_O Q$.

2. States and Actions for Industrial Discharges

In this section, we will introduce a finite state machine which will define a transition function in the domain of *industrial discharges*. It is noteworthy to mention the work of Gimeno *et al.* in [8] and the work of King *et al.* in [10] *w.r.t* . the construction of finite state machines to model the cause-effect relationships at the WWTP level under the operation of an activated sludge system. In order to motivate our finite state machine, that models a larger scale (*i.e.* the UWS scale), let us consider the following scenario:

> Suppose that the industry *La Clarita* dedicated to the production of yoghurts faces a problem in the production system, and the lactic acid bacteria producing medium needs to be replaced. This implies a complete breakdown in the production, the cleaning and disinfection of all tanks with the consequent washout of the lactic acid producing bacteria, together with the current production of yoghurt. While common emissions from the diary industry are biodegradable, this situation will imply a considerable amount of wastewater with high content of organic matter, fats and greases from the milk, as well as a low pH due to the lactic acid bacteria medium.

This is a common situation causing a disturbance in the UWS (*i.e.* , the sewer system, the WWTP and the stretch of the river that finally receives the treated wastewater). In order to *model* and *generalize* the situation described above, let us consider Figure 1. Figure 1 shows a diagram where each node represents a possible situation in which the system can be found and the arrows represent a transition, that is, a possible management action. This global automata can be instantiated according to a particular situation, for example, the one described above.

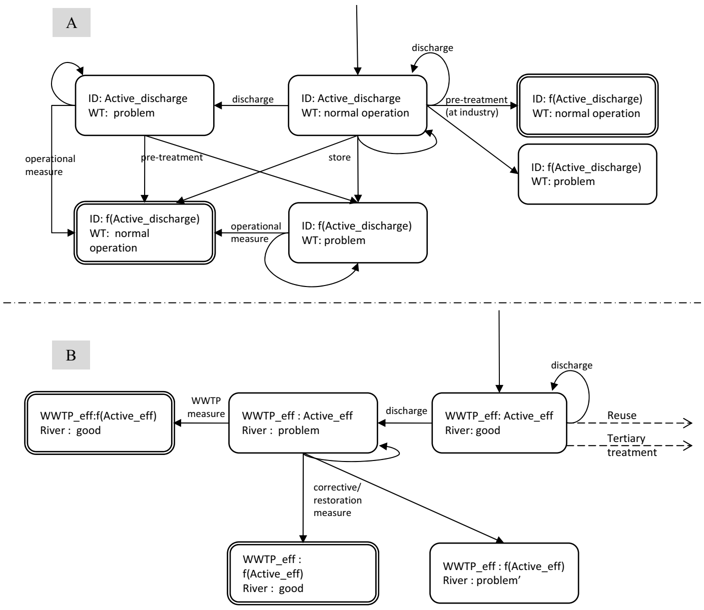

Figure 1. An automata of finite states for considering **A**: problems at *activated sludge* municipal WWTPs given an industrial discharge and **B**: problems at rivers given a WWTP effluent

The states presented in this diagram have been described using the following relevant aspects in relation to the case presented and with the purpose to simplify the possible system situations:

- Industrial discharge wastewater-related aspects (*ID*), according to the amount and type of pollution. We will denote the industrial discharge types as *(ID(Active_discharge))* where *Active_discharge* can be *pH*, *Biochemical Oxygen Demand (BOD*$_5$*)*, *Total Suspended Solids (TSS)*, *etc*.
- WWTP operational situation (*WT*) denoting the sate of the WWTP. We will represent this factor as (*WT(normal_operation)*) and (*WT(Active_problem)*), where *Active_problem* can be any present *biological_operational_problem e.g.* filamentous bulking, dispersed growth, *etc*., or *non_bio_operational_problem e.g.* engine breakdown.
- WWTP effluent characteristics (*WWTP_eff*) denoting a type of treated effluent. It will be denoted as *(WWTP_eff(Active_eff))* where *Active_eff* can be *nutrient_polluted*, *organic_polluted*, *toxic*, *etc*. The classification of specific effluent

discharges threat allows the municipality and the operator of the WWTP to define different policies to prevent risks [3].

- River state (*river*) that denotes *good* state or some *problem* in the river. It will be represented as *(river(good))* and *(river(problem))* where *problem* can be *eutrophication*, *oxygen depletion*, *etc*.

Note that the function *f(Active_discharge)* expresses a transformation process, that is, indicates a change in the value of *ID*. So, it supposes the existence of functions and/or models to express the transformation of pollutants, substances, *etc*. As a consequence *ID(Active_discharge)* can result, after an action, in *ID(f(Active_discharge))*. In the same way, the consideration of a new function *g* is going to be needed when considering specifically the *pre_treatment* transition, which will be dependent on the type of discharge as well as the type of the specific problem needed to be tackled. Accordingly, *ID(Active_discharge)* will result in *ID(g(Active_discharge, Active_problem))*, supposing the existence of a meta-model or a function dealing with the specific pre-treatment needed in each case.

It is important to emphasize that the criteria used when selecting the relevant parameters that describe the machine states are twofold:

- to enlarge the scale modelled over the WWTP level taking into account some states before and after *i.e.* the *ID()* and the *river()*, respectively. This, permits to represent some of the complex dynamics involved within UWS; for instance, the fact that industrial wastewater discharges affect the treatment process and in consequence the quality of the river that receives the treated wastewater.
- to simplify the system by evading the consideration of feedbacks but keeping the most relevant characteristics without loosing detail on the cause-effect relationships between the main components of the UWS.

3. Representing Actions and Reasoning About Actions

In this section, we define a (partial) transition function for managing industrial discharges. This transition function will be captured by a set of effect propositions (this means effect propositions of the form (1)) and the finite state machines of Figure 1.

We are going to start by considering the set of effect propositions related to Figure 1-A (this set of effect propositions will be denoted by Π_{wt}):

discharge **cause** $wt(problem(Active\_discharge)) \vee wt(normal\_operation)$ **if** $id(Active\_discharge),$
$wt(normal\_operation).$

pre_treatment_industry **cause** $wt(normal\_operation) \vee wt(problem(Active\_discharge))$
if $id(Active\_discharge), wt(normal\_operation).$

pre_treatment_industry **cause** $id(f(Active\_discharge))$ **if** $id(Active\_discharge), wt(normal\_operation).$

store **cause** $wt(normal\_operation) \vee wt(problem(Active\_discharge))$ **if** $id(Active\_discharge),$
$wt(normal\_operation).$

store **cause** $id(f(Active\_discharge))$ **if** $id(Active\_discharge), wt(normal\_operation).$

pre_treatment **cause** $wt(normal\_operation) \vee wt(problem(Active\_discharge))$
if $id(Active\_discharge), wt(Active\_problem).$

pre_treatment **cause** $id(g(Active\_discharge, Active\_problem))$ **if** $id(Active\_discharge),$
$wt(Active\_problem).$

operational_measure **cause** $wt(normal\_operation) \vee wt(problem(Active\_discharge))$
if $id(Active\_discharge), wt(Active\_problem).$

Now, we are going to present the set of effect proposition related to Figure 1-B (this set of effect propositions will be denoted by Π_{river}):

discharge **cause** $river(good) \vee river(problem(Active\_eff))$ **if** $WWTP\_eff(Active\_eff), river(good).$

WWTP_measure **cause** $river(good) \vee river(problem(Active\_eff))$ **if** $WWTP\_eff(Active\_eff),$
$river(problem).$

WWTP_measure **cause** $WWTP\_eff(f(Active\_eff))$ **if** $WWTP\_eff(Active\_eff), river(problem).$

restoratoin_measure **cause** $river(good) \vee river(problem(Active\_eff))$ **if** $WWTP\_eff(Active\_eff),$
$river(Active\_problem).$

restoratoin_measure **cause** $WWTP\_eff(f(Active\_eff))$ **if** $WWTP\_eff(Active\_eff), river(problem).$

Observe that it is easy to prove that both Π_{wt} and Π_{river} define a *consistent* transition function, *i.e.* , $\Phi_{\Pi_{wt}}$ and $\Phi_{\Pi_{river}}$. In fact, $\Pi_{wt} \cup \Pi_{river}$ also defines a *consistent* transition function $\Phi_{\Pi_{wt} \cup \Pi_{river}}$.

We are going to introduce a set of observation propositions which are related to the scenario described in §2 (this set of observation propositions are denoted by $O_{discharge}$):

initially $id(bod5\_very\_high\_AND\_pH\_very\_low).$
initially $wt(normal\_operation).$

Once we have specified the finite state machines introduced in Figure 1 in terms of effect propositions and observations propositions, we can use the above formulation to enable different kinds of reasoning about actions such as predicting the future from information about the initial state, assimilating observations to deduce about the initial state, combination of both and planning.

3.1. Temporal Projections

In temporal projections, the observations are only of the form (4) and the only interest is to make conclusions about (hypothetical) future. For instance, let us consider Π_{wt} and $O_{discharge}$. We can see that there is just one initial state corresponding to $(\Pi_{wt}, O_{discharge})$: $\sigma_0 = \{id(bod5\_very\_high\_AND\_pH\_very\_low), wt(normal\_operation)\}$ Let us suppose that $problem(id(bod5\_very\_high\_AND\_pH\_very\_low)) = filamentous\_bulking$. Now, let us consider the query Q_1:

$$wt(normal\_operation) \textbf{ after } discharge, pre\_treatment$$

We can see that $wt(normal\_operation)$ holds in the state $[pre\_treatment, discharge]\sigma_0$. This means that $\Pi_{wt} \models_{O_{discharge}} Q_1$; hence, the industry *La Clarita* can perform its discharge if a pre-treatment is applied after its discharge. Observe that, if we consider the query Q_2 $wt(normal\_operation)$ **after** $discharge$, it is false that $wt(normal\_operation)$ holds in the state $[discharge]\sigma_0$. This means that $\Pi_{wt} \not\models_{O_{discharge}} Q_2$; therefore, the industry *La Clarita* cannot perform its discharge if a pre-treatment is not applied.

3.2. Reasoning About the Initial Situation

In reasoning about the initial situation, the observations can be about any situation, but the queries are only about the initial situation. For instance, let O' be:

$$O' = \{\textbf{initially } wt(normal\_operation); wt(filamentous\_bulking) \textbf{ after } discharge\}.$$

This means that we only know that before the discharge was done the WWTP had a normal operation and after the discharge was done the WWTP has the problem of filamentous bulking. We can see that (Π_{wt}, O') has just one initial state $\{wt(normal\_operation), id(bod5\_very\_high\_AND\_pH\_very\_low)\}$. This means that $\Pi_{wt} \models_{O'} id(bod5\_very\_high\_AND\_pH\_very\_low)$. Therefore, from the observation O' we can backward and conclude that the discharge had *bod5 very high* and *pH very low*.

Following this query and by instantiating the automata of Figure 1-B using the results of the first instantiated automata (Figure 1-A), we can proceed to reason about the final river state. Accordingly, let O' be:

$$O' = \{\textbf{initially } wttp\_eff(organic\_polluted); river(oxygen\_depletion) \textbf{ after } discharge\}.$$

This means that before discharging to the river the WWTP that had overcame the problem of filamentous bulking will, accordingly, have an organic polluted effluent since part of the activated sludge overflows. (Π_{river}, O') has initial state $\{wwtp\_eff(organic\_polluted), river(good)\}$ which means that $\Pi_{river} \models_{O'} river(good)$. We can go back to the observation O' and conclude that the discharge of an organic polluted effluent from a WWTP to the river cause the depletion of oxygen into the river.

Let us exemplify another query in the same scenario but from a different initial state (*e.g.* the same industrial discharge but knowing that the load of nitrogen in the wastewater is very low compared with the high load of BOD_5). Hence, the observation propositions related to this new query denoted by $O_{discharge}$ are:

initially $id(bod5\_very\_high\_AND\_nitrogen\_low)$.
initially $wt(normal\_operation)$.

Let $O' = \{$**initially** $wt(normal\_operation); wt(filamentous\_bulking)$ **after** $discharge\}$. This means that, in the same line as the first query, we only know that before the discharge was done the WWTP had a normal operation and after the discharge was done the WWTP has the problem of filamentous bulking. We can see that (Π_{wt}, O') has just one initial state $\{wt(normal\_operation), id(bod5\_very\_high\_AND\_nitrogen\_low)\}$ which means that $\Pi_{wt} \models_{O'} id(bod5\_very\_high\_AND\_nitrogen\_low)$. Therefore, from the observation O' we can backward and conclude that the discharge had *bod5 very high* and *nitrogen low*. Consequently, the origin of the same *Active_problem i.e. filamentous bulking* at the WWTP has been diagnosed and the subsequent corrective action or plan of actions can be better assessed, and the solution of the problem can be tackled within a more purposeful way, since the causes are specifically diagnosed.

Let us now consider a third query in which the initial propositions are:

initially $id(bod5\_very\_high\_AND\_nitrogen\_low)$. **initially** $wt(filamentous\_bulking)$.

In this query let O' be:

$O' = \{$**initially** $wt(filamentous\_bulking); wt(normal\_operation)$ **after** $operational\_measure\}$.

This means that it is known that before an operational measure was performed the WWTP has the problem of filamentous bulking. Moreover, (Π_{wt}, O') has the following initial state $\{wt(filamentous\_bulking), id(bod5\_very\_high\_AND\_nitrogen\_low)\}$ which means that $\Pi_{wt} \models_{O'} id(bod5\_very\_high\_AND\_nitrogen\_low)$. Consequently, from the observation O' we can backward and conclude that the discharge had *bod5 very high* and *nitrogen low*. Stating these observations as the causes of the *Active_problem i.e.* filamentous bulking, the operational measure can then be focused on dealing with them an not with other possible causes of filamentous bulking.

So far, we have showed that by considering a formulation of transition functions in $\mathcal{A}$ one can perform reasoning at a high level of sequences of action in the domain of industrial discharges. An important feature of the formulation that we have introduced is that it can be transformed into an answer set program [1]. The detailed presentation of the transformation of $\Phi_{\Pi_{wt}}$ and $\Phi_{\Pi_{river}}$ into an answer set program is out of the scope of this paper; however, the interested reader can find at *http://lequia.udg.cat/members/maulinas/CCIA09-program.txt* a small implementation of our wastewater management scenario.

4. Conclusions

In this paper, we propose to use recent developments in knowledge representation languages and non-monotonic reasoning methodologies for representing and reasoning about actions for the Management of Urban Wastewater Systems. To this end, we have presented an abstraction of sequences of actions in industrial discharges in terms of finite state machines in order to define a transition function able to capture transitions between states and actions to states (§2). Observe that the definition of transition functions in domains as UWS requires a deep understanding of the domain. However, the definition of these transition functions is a good approximation for performing adequate design of decision-making systems. We have showed that by considering a formalization of our finite state machines into the action language $\mathcal{A}$, one is able to perform temporal projections about the effect of a sequence of actions (§3.1). Also, we are able to perform backward reasoning in order to explain a given state in an UWS.

Even though, in this paper we have only discussed the formulation of transition functions in terms of $\mathcal{A}$, we want to point out that this formulation can be computed via answer set solvers. This suggests that one can consider declarative approaches as answer set programming for performing reasoning about actions in general and in this case for UWS. This approach, among other advantages, allows an easier and friendly way of up-dating the knowledge base domain. The use of declarative action languages such $\mathcal{A}$ requires less programming skills and is more understandable for non computer scientists experts. Moreover, the interaction between the final user (*e.g.* water managers) and the knowledge-based system can be easily performed through the use of *queries*.

Acknowledgement

We are grateful to anonymous referees for their useful comments. We would like to acknowledge support from the EC founded project ALIVE (FP7-IST-215890). The views expressed in this paper are not necessarily those of the ALIVE consortium.

References

[1] C. Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, Cambridge, 2003.

[2] C. Baral, K. Chancellor, N. Tran, N. Tran, A. M. Joy, and M. E. Berens. A knowledge based approach for representing and reasoning about signaling networks. In *ISMB/ECCB (Supplement of Bioinformatics)*, pages 15–22, 2004.

[3] B. Crabtree and G. Morris. Effective environmental regulation to maximise the benefits of integrated wastewater management. *Water Science & Technology*, 45(3):211–8, 2002.

[4] S. Dworschak, S. Grell, V. Nikiforova, T. Schaub, and J. Selbig. Modeling Biological Networks by Action Languages via Answer Set Programming. *Constraints*, 13(1):21–65, 2008.

[5] T. Eiter, W. Faber, N. Leone, G. Pfeifer, and A. Polleres. A logic programming approach to knowledge-state planning, ii: The DLV^{K} system. *Artif. Intell.*, 144(1-2):157–211, 2003.

[6] M. Gelfond and V. Lifschitz. The Stable Model Semantics for Logic Programming. In R. Kowalski and K. Bowen, editors, *5th Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.

[7] M. Gelfond and V. Lifschitz. Representing Action and Change by logic programs. *Journal of Logic Programming*, 17(2,3,4):301, 323 1993.

[8] J. Gimeno, J. Béjar, U. Sànchez-Marrè, U. Cortés, I. Rodríguez-Roda, M. Poch, and J. Lafuente. Providing wastewater treatment plants with predictive knowledge based on transition networks. In *Int. Conference on Intelligent Information Systems (IIS'97)*, pages 355–359. IEEE Computer Society Press. ISBN 0-8186-8218-3. Freeport, The Bahamas, 1997.

[9] H. Grüttner. Regulating industrial wastewater discharged to public wastewater treatment plants-a conceptual approach. *Water Science & Technolgy*, 36(2):25–33, 1997.

[10] R. King, A. Stathaki, F. Koumboulis, and N. Kouvakas. An intelligent automaton for the control of an active sludge process. In *International Conference on Industrial Technology*, volume 2. IEEE Computer Society Press. ISBN 0-7803-7852-0, 2003.

[11] S. Rusell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Pretice Hall Series in Artificial Intelligence, 2003.

[12] M. Sànchez-Marrè, K. Gilbert, R. Sojda, J. Steyer, P. Struss, I. Rodríguez-Roda, J. Comas, V. Brilhante, and E. Roehl. Intelligent Environmental Decision Support Systems. In A. Jakeman, editor, *Environmental Modellilng, software and decision support: state of the art and new perspectives*, number 3 in Developments in Integrated Environmental Assessment, chapter VI, pages 119–144. Elsevier, The Netherlands, first edition, 2009.

[13] S. SMODELS. Helsinki University of Technology. http://www.tcs.hut.fi/Software/smodels/, 1995.