Jul 1st, 12:00 AM

A multi-agent framework for an IEDSS in urban water management

Joaquín Izquierdo

Idel Montalvo

Rafael Pérez-García

Joanna A. Gutiérrez-Pérez

International Environmental Modelling and Software Society (iEMSs)
2010 International Congress on Environmental Modelling and Software
Modelling for Environment's Sake, Fifth Biennial Meeting, Ottawa, Canada
David A. Swayne, Wanhong Yang, A. A. Voinov, A. Rizzoli, T. Filatova (Eds.)
http://www.iemss.org/iemss2010/index.php?n=Main.Proceedings

A multi-agent framework for an IEDSS in urban water management

Joaquín Izquierdo, Idel Montalvo, Rafael Pérez-García, Joanna A. Gutiérrez-Pérez
Instituto de Matemática Multidisciplinar, Universidad Politécnica de Valencia
Camino de Vera s/n, 46022, Valencia, Spain
{jizquier,imontalvo,rperez,joagupre}@gmmf.upv.es

Abstract: Managers of water supply companies must be concerned about the lack of integrity of their systems. The consequences are varied: service disruptions; pipe breaks causing water wastage; pipe cracks causing water leaks and pathogen intrusion, a serious human health risk. Despite its importance, water hammer, one of the main vulnerabilities of water systems, still represents a challenge, due to the high level of complexity this problem exhibits. It is a fact that decision making regarding this issue lack efficiency, since more powerful tools are needed. One of the drawbacks of computer packages for hydraulic transient simulation in complex systems resides in modelling complex devices encountered in real installations. Computer packages in the market model independent devices with accuracy. Yet, defining compound devices has proven problematic. The used trends to model complex devices are twofold. Either, to define each combination of simple elements as a new (simple, even though much more complex) element; or to link different simple elements by short pipes. The former produces multiplication of code, which renders packages inefficient. The latter is burdened by the Courant condition – a necessary condition for stability – that turns calculations non affordable both on grounds of time and computational resources. In this paper, a new platform to simulate transient phenomena in pipe networks, DIAGAST.ING, is presented. It uses a multi-agent approach to efficiently overcome the mentioned problems. Last but not least, a multi-agent-based system favours the parallelization of the calculation algorithm. We claim it is an efficient DSS in urban water management.

***Keywords**:* Hydraulic transients, water distribution networks, vulnerability, multi-agent systems, decision support.

1. INTRODUCTION

A hydraulic network is a really complex distributed system (see, e.g. Izquierdo and Iglesias [2004]). It is constituted by an interconnected and intricate set of pipes that also includes certain elements of complex behaviour. And, since its main purpose is to provide citizens with a service of first necessity, there is no room for doubt about its strategic, social, environmental and sanitary importance.

Managers of water supply companies must be concerned about the lack of integrity of their systems due to a number of reasons. Failure consequences are manifold. Service irregularities, sometimes acceptable, may end up as severe service disruptions. Pipe breaks may cause big investment losses, water wastage, and third party damages involving different economic losses. Cracks in pipes may cause two of the most insidious effects of environmental concern in urban water management: water leaks, which represent a continuous and imperceptible, but too large, waste of water, and pathogen intrusion, which impairs water quality and represents a serious risk for human health.

The elements that integrate a hydraulic network (nodes, tanks and reservoirs, pumps, different control devices, etc.) have specific behaviours, not necessarily autonomous, and interact with each other, partly in a predefined way, in accordance with appropriate design considerations. However, the appearance in the network of states far from the design point (due to abrupt changes in the demands, fire events, etc.) makes that the behaviour of such elements is also conditioned by certain successions of events.

In spite of its importance, nevertheless, water hammer still represents a challenge for water companies. At the same time, powerful tools for decision making regarding this issue are necessary. One of the most severe drawbacks of conventional computer packages devoted to the simulation of hydraulic transients in complex systems comes from the difficulties for modelling (programming) complex devices encountered in real installations. There are many computer packages in the market that model independent devices with great efficiency. Yet, the definition of compound devices has proven problematic. The used trends to model complex devices so far are twofold. Either, to define each combination of simple elements as a new (simple, even though much more complex) element; or to link different simple elements by short pipes. The former produces open multiplication of code, which renders packages inefficient. The latter is clearly burdened by a well-known constraint: the Courant condition – a necessary condition for the method stability – that takes calculations to non affordable situations both on grounds of time and computational resources.

It is a very well-known fact that the transmission of perturbations is a phenomenon of transmission of information, that is to say, a communication phenomenon. In this process, the elements that get involved possess individual behaviour rules that can be influenced or modified by the behaviours of other elements. In turn, a fundamental part of the phenomenon is the succession along the time of questions and answers, that is to say, exchange of information, among the different elements of the system. That permanent dialogue among such elements characterizes the events that take place.

In effect, given the information corresponding to an initial condition, it is possible to build its future evolution by simply transmitting this information through the characteristics. We refer to the method of characteristics (MOC), the most utilized numeric method for the solution of the hyperbolic equations describing the process (Abreu et al. [1995]). In the MOC space and time are discretized. This generates a discrete set of calculation points that are the active elements owners and carriers of certain information that changes to defined impulses with time. The time discretization sets the schedule of the combined activity. This information is transmitted to other elements on request. The obtained information allows the different elements to complete their information, define their behaviour and elaborate answers that, in turn, feed other elements.

The emergent behaviour of the calculation points of a pipe allows to see the pipe as an element of another level whose complete behaviour still needs of new rules to communicate with the rest of elements of the network. Those other elements from the network to which pipes are connected are known with the general name of boundary conditions. The behaviour of they all are mutually influenced. As a consequence, it is essential to define not only all the potential elements but also their possible interrelations.

The more common elements in a network are the consumption nodes or junctions. The activity of a consumption node typically consists in the negotiation of the characteristics of the pipes joining at the node (see a detailed description in Izquierdo and Iglesias [2002, 2004] and in Abreu et al. [1995]), the elaboration of the own information that determines its behaviour and, in turn, the dissemination in an appropriate way of specific information to the connected pipes. Neither from the conceptual point of view, nor from the physical one, nor from that of programming there exists substantial differences with the behaviour of the inner nodes of a pipe, except for the fact that a consumption node can have, as suggested by its name, an associate consumption, and because in a consumption node more than two pipes may join, that is to say, it possibly has to manipulate more than two characteristics.

Other boundary conditions, on the other hand, are more complex. They are specific elements with specific functions. They are these varied devices and combinations of them found in hydraulic installations that produce, damp, amplify, control, etc. the perturbations. The individual devices (pumps, valves, air vessels, etc.) are described in a satisfactory way in the literature using different types of models; in general, they are steady state models or lump models, Abreu et al. [1995], Izquierdo and Iglesias [2002]. However, most current commercial computer packages in the market and the codes of domestic character devoted to the simulation of hydraulic transients in complex systems are strongly biased by diverse problems. One of the most serious, if not the most one, comes from the difficulty in modelling (programming) compound boundary conditions that appear in actual facilities. The definition of such compound devices represents a crucial point that is able to pinpoint the difference between efficient and non-efficient codes, Izquierdo and Iglesias [2004].

In essence, there are two trends to model such complex devices. One consists in defining combinations of simple elements as new elements, which we will continue calling simple in the sense that they are defined by a unique routine and are accessible by means of their own icons in the computational environment; it is the case of DYAGATS, Izquierdo et al. [1996], among others. The other considers different simple elements linked through short pipe sections. With the first alternative the codes of the elements, after undergoing certain modifications not necessarily straightforward, proliferate in different routines (an entire fauna of icons appears), what makes programming inefficient and obsolete. With the second, one incurs in full in the so-called curse of the short pipe, a very well-known fact, Abreu et al. [1995]: the Courant-Friedricks-Lewy condition, which is necessary to guarantee stability, takes the calculations to inadmissible situations in terms of computational resources (time and memory). There exist codes in the market that, in a hidden way for the user, simply assign a minimum length to the shortest pipe, to obviate such a circumstance. Other alternatives have been described in the literature, as the one using so-called rigid pipes to model such short reaches (see Abreu et al. [1995]). In any case, they are approximations to the problem that use simplifications that are always debatable.

In this contribution we present a computer package for the simulation of transients in hydraulic networks, called DIAGAST.ING, which uses multi-agent techniques. DIAGAST is the acronym in Spanish of 'intelligent design against water hammer in pipe systems'; ING refers to the trademark IngeniousWare©. On one hand, the whole underlying philosophy in the simulation of transients is coherent, since it includes in the category of agents any element that has a defined behaviour and that interacts with other elements. Also, the above mentioned problems in the definition of compound boundary conditions are obviated in an elegant and efficient way. Of course, simple elements are defined in an appropriate way (as in DYAGATS, for example) that, at the same time, is unique. That is to say, there is no code repetition at all. And, by using a multi-agent approach, the combinations of different simple elements in one location are carried out through the introduction of new agents, called facilitators, that harmonize the traffic of questions and answers among the simple elements. In other words, facilitators are the moderators of the dialogue among them. However, the biggest advantage in the approach that constitutes the base of DIAGAST.ING is that a system based on agents allows the parallelization of the calculation algorithm, what favours a better use of computer resources. We claim that DIAGAST.ING constitutes a DSS for urban management that is clearly superior to other similar codes in the market.

2. MULTI-AGENT PARADIGM

In the study of complex systems, computer programs have played an important role. However, the actual process of writing software is a complicated technical task with much room for error. The multi-agent philosophy adopts a modelling formalism based on a collection of independent agents interacting through discrete events. Simulation of discrete interactions between agents stands in contrast to continuous system simulations, where simulated phenomena are quantities in a system of coupled equations.

An agent is any actor in a system: any entity that can generate events that affect itself and other agents. In the architecture we consider here, agents are semi-autonomous processing elements working together to solve a real complex problem in urban water management. That is to say, it is a framework for cooperative distributed problem solution that divides the problem, distributes the various sub-problems, synthesises the results, and optimizes the solution through coherence and coordination.

In the problem we consider here, agents are pipe discretization points, consumption nodes, connecting pipes, supply sources, different devices, ground patches containing the network; as well as district metered areas, which are set of nodes, pipes, sources, and patches. Even the whole network is an agent following specific scheduled actions. In these last two cases, agent behaviour is defined by the emergent actions of the agents they contain. Agents define the basic objects in the system – the simulated components. The simulation occurs in the modelled world itself, and it is frequent to speak of agents as living in an environment, which, in its turn, can be an agent itself.

Besides its attributes, an agent is characterized by a number of static and dynamic variables whose values describe the agent's state at any given time. Using these variables, the system can simulate the evolution of the agent's dynamic states and trigger the relevant objectives.

An agent is also associated with a behaviour which is represented by a set of multi-layered directed graphs, each one composed of a set of objectives for the agent to reach (see Figure 1, showing the facilitator's behaviour described in the next section, as an example). Behavioural graphs are data structures used to define complex agent behaviours. These graphs are made out of nodes representing objectives. The objectives are structured in a hierarchical way such that elementary objectives are associated with actions the agent can execute. Different objectives are connected by links that may be simple of multiplexers, depending on the number of transmitted signals. In general, agents own sets of objectives, which can be either simple or composed, matching their needs, and objectives are associated with rules describing the activation, execution and completion of the objective. Activation rules, closely associated with the different links, are used to influence the state of a potential descendent objective. Execution rules control the execution of an objective and modify the agent's state. Finally, completion rules use the state values of the objective to determine the action to be carried out. To accomplish its objectives, an agent has a specific pool of resources.

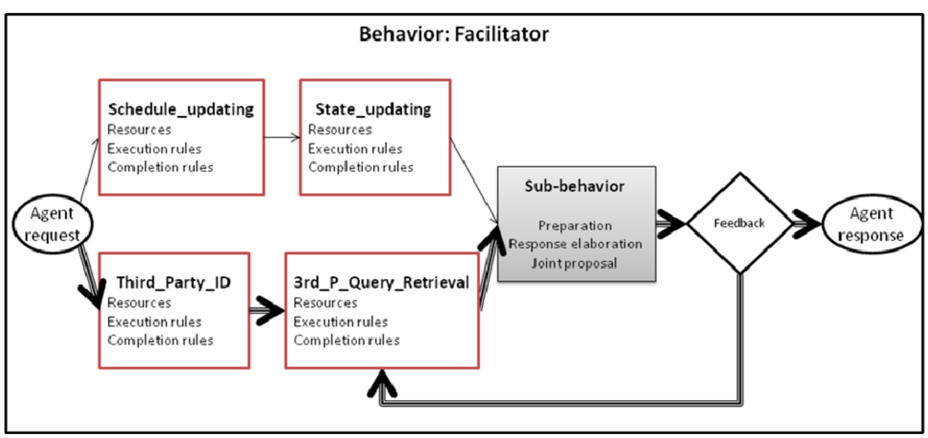

Figure 1. Example of agent behaviour

The way rules apply depends on time, agent's state, and environment's state. An agent selects its current objective according to the graph structure, to previously executed objectives, and to priorities regarding the abovementioned rules. As can be observed in Figure 1, the structure of the multi-layered graph may contain behaviours at different levels of abstraction. Compound objectives can be devised as decomposable structures representing sub-behaviours, with similar structure.

Once agents have been defined and their relationships established, a schedule of discrete events defines a process occurring over time. Individual actions take place at some specific time, which advances alongside events scheduled at successive times. A schedule is another data structure that combines actions in the specific order in which they should execute. The passage of time is modelled by the execution of the events in some sequence. Instructions are given to hundreds or thousands of independently operating agents. This makes it possible to explore the connection between the micro-level behaviour of individuals; and the macro-level patterns that emerge from the interaction of many individuals.

A final step consists in observing the model and recording what is happening. Observers perform these actions. In most platforms there are also agents with specific tasks, such as plotting, storing data, monitoring and displaying certain variables, etc.

The trend in recent years to include multi-agent techniques as an interesting alternative for solving complex problems is clear from the following examples (restricted to the water field, the authors' field of expertise): Izquierdo et al. [2008] on multi-agent applications in urban hydraulics; Maturana et al. [2006] on water and waste water control system architecture; Kotina et al. [2006] on control systems for municipal water; Nichita and Oprea [2007] on water pollution diagnosis; Feuillette et al. [2003] on water demand management for a free access water table; Hai-bo et al. [2005] on water quality; Becu et al. [2001] on water management at catchment scale; Cao et al. [2007] on optimization of water networks; Mikulecký et al. [2008] on water management at river basin scale; Hailu and Thoyer [2005] on allocation of scarce water; and Herrera et al. [2009] about division of water supply networks into district metered areas, among others.

3. THE PLATFORM DIAGAST.ING

The platform of DIAGAST.ING is the world or environment where agents participating in the simulation of hydraulic transients in complex systems live in. Besides the menu and the toolbars, four frames can be observed in figure 2. The left lower one is the drawing frame, and it contains the installation plan. The other frames correspond to three selected profiles.

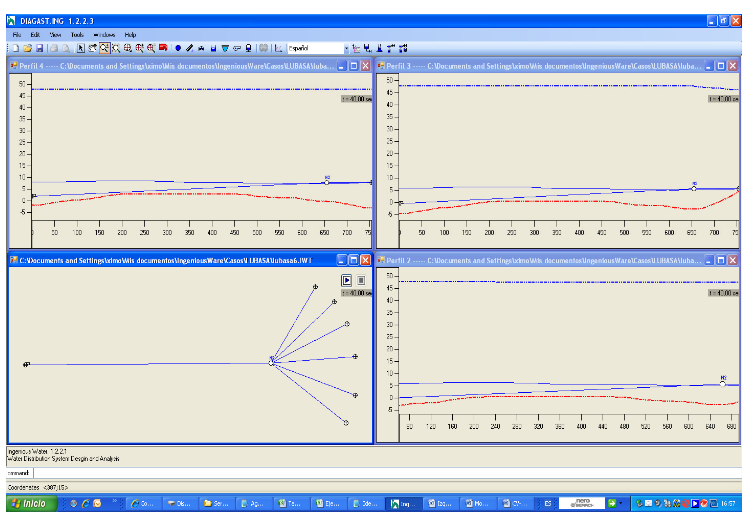

Figure 2. Platform of DIAGAST.ING

As already observed, the breeds of agents that get involved in this platform are, basically, pipe calculation nodes, pipes, consumption nodes, reservoirs, tanks, valves, pumps, air vessels, surge tanks, unidirectional tanks, other simple devices; profiles, monitors, time graphs, mixed space-time graphs, charts of data, the installation itself. Even the user is an agent that interacts with any other agent.

Agents belonging to different breeds are created and incorporated to the platform in an individual or collective way. Some agents are created using the built-in tools that allow locating specific elements at specific coordinates. Other agents are created in an automatic way when certain processes are initialized. For example, the pipe calculation nodes are created when a suitable discretization is defined. The different graphs or outputs of data are created at the user's request, which, in his/her turn, can interact with their properties, etc.

Upon the creation of an agent, it has an associate set of properties defining it. Such properties can be individual, used by the agent in an exclusive way, or participative, used by the agent in its relationship with other agents. These last properties encapsulate the protocols of exchange of information. As a result, during the process, each agent will be able to recognize any other agent approaching it; will know how to dialogue appropriately with this agent, and will be in condition to offer the sought answer.

The MOC itself is a mechanism of transmission of information among certain agents. It is used by the internal nodes of a pipe and also by the pipes themselves to address the consumption nodes and the other elements they are connected to. Their behaviour (the activity that they carry out) establishes a dialogue with other agents, performed by obeying a schedule that is established at the time the number of calculation points and the time step of the discretization are decided. In that dialogue, the agents interrogate other agents, demand certain answers and, in turn, respond to the questions they are posed. This is, accordingly, an example in which the multi-agent methodology accommodates in a perfect way the classic requirements based on the different equations that define the model.

Other agents, such as pumps, tanks/reservoirs, valves, etc., are defined by their so-called constitutive equations in a very well-known way (see, for example, Abreu et al. [1995], Izquierdo and Iglesias [2002, 2004]). They are agents of other breeds with their characteristic variables and perfectly defined behaviours. Some of these agents accommodate their behaviour to the preset common schedule. Others, in turn, use own schedules that need to be synchronized with the global schedule of the time discretization. These individual schedules describe the manoeuvres of such devices accurately.

Finally, other agents exist, with the specific function of facilitating the interconnection and dialogue among different simple agents that constitute a compound boundary condition. These agents, which we call facilitators, are introduced in the next section and we show some examples of their operation.

Since we cannot present DIAGAST.ING architecture in full, we briefly present the specification of facilitators and their behaviours, leaving aside such important aspects as scenarios, groups of agents, accessible databases, and a more detailed description of the IDE (integrated development environment), among others.

4. THE FACILITATOR, A BROKER AGENT

The facilitator is a new class (breed) of agent designed to put in touch different simple devices that can integrate a compound element or general boundary condition in an appropriate way.

It is a pipe-agent that inherits, therefore, all the properties and characteristics of the pipe-agents, interprets them with a personal perspective and incorporates other new characteristics. A facilitator must be first declared as such. Then, in an autonomous way, the agent itself modifies those internal variables necessary to carry out its specific function. As a consequence of its properties, the agent will know perfectly the elements it has to put in touch, what to ask to any of them, how to elaborate the necessary information, and how to respond to each one with the requested information.

The first level of the facilitator's behaviour can be observed in figure 1. Upon being requested, the agent's behaviour follows the first links, which are concurrent and have activation rules that enable the first two objectives, namely, the agent's schedule updating followed by its state updating, and the identification of the third party elements the agent have to put in touch. Following this second thread, the facilitator has either to know or query the needs of the other actors and then execute a lower level behaviour to perform a number of tasks, namely, mediation preparation, response elaboration and joint proposal specification (not shown in the figure). Prior to exit, the necessary feedback is performed to achieve the necessary consensus. The various nodes within behaviours and sub-behaviours use a number of resources and activation and completion rules that jointly define the whole facilitator's behaviour.

We present here two simple instances of facilitators. The first one is the facilitator between a valve and a tank. This agent is a pipe without physical properties of interest that maintains the relationship between the valve and the reservoir by means of a really simple action: it asks the tank for its current water level and the cross section corresponding to this level, and inquires the valve about its opening state, flow and head; then, it elaborates internally those data and returns appropriate information to both devices. The answer for the valve is a (negative or positive) characteristic, which is what the valve waits for. The information given to the tank contains the necessary information for it to modify appropriately its level of water. It is worth observing here that physical and mathematical properties of the phenomenon are permanently present. In an analogous way it is possible to explain the activity of facilitators among other elements.

In figure 3 a pumping facility to three reservoirs through a looped network is presented. The pumping station (leftmost element) is protected by an air vessel close to the pumps. In its turn, one of the reservoirs (at the bottom) is regulated by a nearby valve to its left. While most of the pipes are standard, the pipes linking the pump station with the air vessel and the valve with the reservoir are facilitators. They are drawn, identified, topologically connected, etc., as the other pipes, but have a specific variable declared as true. Since facilitators also know the other elements they are connected to, they accomplish perfectly their mediators' role in the dialogue between the elements they connect.

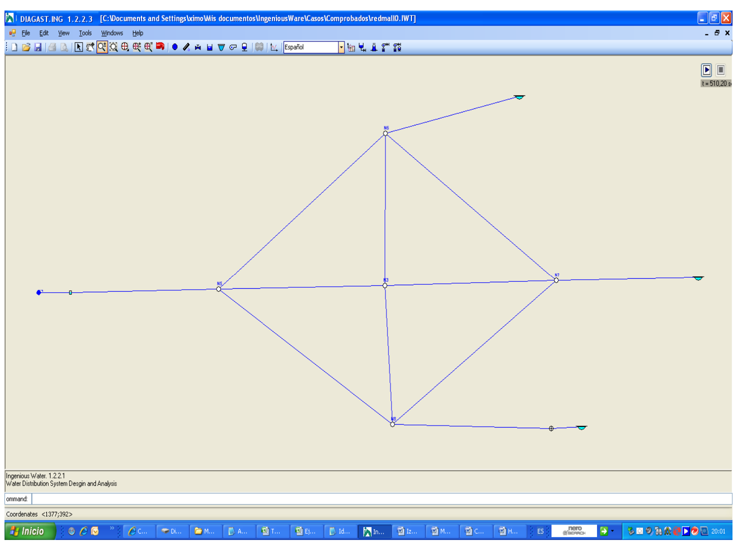

Figure 3. Facilitators between pump and air vessel and between valve and reservoir

As said before, facilitators manage to overcome two important problems affecting similar tools in the market, namely, repetition of code and use of short pipes. The former turns packages obsolete and inefficient in terms of code writing and debugging, and, above all, in terms of extensibility. The latter, has two more pernicious effects. Computer codes using short pipes have two main problems when trying to overcome the Courant condition. Either they simply cannot perform certain hydraulic simulations due to lack of computational resources or, what is worse, produce unrealistic results since they perform some artificial adjustment, which is opaque for the user. In most cases wave speeds are approximated beyond reasonable ranges. But to these authors' knowledge there are codes in the market that perform other even more unjustifiable adjustments such as assigning some minimum length to the pipes that are so short as to pose problems to the Courant condition compliance.

5. CONCLUSIONS

Water supply is one of the more recognizable and important public services contributing to quality of life. Consequently, the security and the integrity of this service must be guaranteed. One of the phenomena that put in danger such a security is that of hydraulic transients. It is a very complex phenomenon, described by complicated models, solved by means of delicate numerical methods, difficult to visualize and to interpret, and not inclined to straightforward and simple judgment and to decision-making. The computational implementation of the methodologies used to solve this problem needs computer intensive use. Currently, the calculation power and the different capabilities of the computers are enough to succeed in modelling such a phenomenon. Nevertheless, it is still necessary to consider certain aspects directly derived from the characteristics of the models that issue important warns about such implementations.

In this paper we have presented a new computer package, DIAGAST.ING, based on a multi-agent technology that approaches the problem in an efficient way, since it is able to obviate certain restrictive implementation drawbacks. On one hand, it avoids in an elegant way the problems in the definition of compound boundary conditions. This is translated in that code repetition does not exist and short pipes are avoided. Indeed, by means of the proposed multi-agent approach, the combinations of different simple elements in only one location are carried out through new agents, called facilitators, that moderate the traffic of questions and answers among simple elements. That is to say, facilitators harmonize the dialogue among them. However, the biggest advantage of the approach that constitutes the base of DIAGAST.ING is that a system based on agents favours the parallelization of the calculation algorithm what leads to a better use of the computer resources.

The presented package constitutes a tool of great interest for water supply managers, for whom transient analysis still represents a pending subject. The use of a complete simulation tool as DIAGAST.ING will allow them to solve the two following problems: avoid under-protection of their systems due to faulty and/or excessively simplified analysis, Izquierdo et al. [2009], and avoid overprotection due to adherence to excessively tight margins of safety, with the extra investment that this implies, for not having a sufficiently powerful tool. Due to these features, DIAGAST.ING may be viewed as an efficient DSS in urban hydraulics, which clearly overcomes other similar tools currently available in the market.

Among the ongoing research lines we have to mention the necessary upgrading of the agent-facilitator that guarantees a perfect operation with any (reasonable) composition of elements in a water distribution network. Also, the parallelization of the algorithm is one of the tasks to face immediately, since it will improve clearly some of the procedures of the package. Currently, agents execute their behaviours in groups assigned to different code threads. Finally, distribution should also be addressed. Although currently the platform is implemented in only one computer, a real distribution of the different tasks can be performed in a straightforward way. It will happen as soon as the first water company show interest in the prototype.

ACKNOWLEDGMENTS

This work has been developed with the support of the project IDAWAS, DPI2009-11591, of the Ministerio de Ciencia e Innovación (Spain) and the scholarship MAEC-AECI 0000202066 granted to the second author by the Ministerio de Asuntos Exteriores y Cooperación of Spain.

REFERENCES

Abreu, J.M., Guarga, R., Izquierdo, J. (Eds.), *Transitorios y oscilaciones en sistemas hidráulicos a presión*, Universidad de Coimbra, Universidad Politécnica de Valencia y Universidad de la República Oriental del Uruguay, 1995.

Becu, N., Perez, P., Walker, A., Barreteau, O., CatchScape: An integrated multi-agent model for simulating water management at the catchment scale, a northern Thailand case study. In: *Ghassemi, F. et al. (Eds). Integrating Models for Natural Resources Management Across Disciplines, Issues and Scales. International Congress on Modelling and Simulation*, Canberra, Australia, 1141-1146, 2001.

Cao, K., Feng, X., Ma, H., Pinch multi-agent genetic algorithm for optimizing water-using networks. *Computers & Chemical Engineering, 31*(12), 1565-1575, 2007.

Feuillette S., Bousquet F., Le Goulven P., SINUSE: a multi-agent model to negotiate water demand management on a free access water table. *Environmental Modelling and Software, 18*(5), 413-427, 2003.

Hai-bo, L., Guo-chang, G., Jing, S., Yan, F., Multi-agent immune recognition of water mine model. *Journal of Marine Science and Application, 4*(2), 44-49, 2005.

Hailu, A., Thoyer, S., Multi-Unit Auctions to Allocate Water Scarcity Simulating Bidding Behaviour with Agent Based Models. *LAMETA* 2005-01, EconWPA, 2005.

Herrera, M., Izquierdo, J., Montalvo, I., Pérez-García, R., Sectorización de redes de distribución de agua basada en técnicas multiagente. IX SEREA'09 Planificación, proyecto y operación de redes de abastecimiento de agua, Valencia, España, 2009.

Izquierdo, J., Iglesias, P.L., Cabrera, E., DYAGATS - Simulación mediante ordenador personal de transitorios hidráulicos en sistemas simples. VII Encontro nacional de Saneamento Basico, Coimbra, Portugal, 1996.

Izquierdo, J., Iglesias, P.L., Mathematical Modeling of Hydraulic Transients in Simple Systems, *Mathematical and Computer Modeling 35*, 801-812, 2002.

Izquierdo, J., Iglesias, P.L., Mathematical Modeling of Hydraulic Transients in Complex Systems, *Mathematical and Computer Modeling 39*(4-5), 529-540, 2004.

Izquierdo, J., Montalvo, I., Pérez, R., Aplicaciones de la inteligencia colectiva (multiagente) a la optimización de procesos en hidráulica urbana. Invited talk at VIII Seminario Iberoamericano – SEREA Influencia sobre el Cambio Climático, la eficiencia energética, de operaciones y Sistemas de Seguridad en el abastecimiento y el drenaje urbano, Lisboa, Portugal, 2008.

Izquierdo, J., Montalvo, I., Pérez-García, R., Alonso, C.D., Sobre ciertas simplificaciones en la simulación de transitorios hidráulicos, IX SEREA'09 Planificación, proyecto y operación de redes de abastecimiento de agua, Valencia, Spain, 2009.

Kotina, R., Maturana, F.P., Carnahan, D., Multi-agent control system for a municipal water system. Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, 464-469, 2006.

Maturana, F.P., Kotina, R., Staron, R., Tichý, P., Vrba, P., 2006. Agent-based water\waste water control system architecture. IADIS International Conference Applied Computing.

Mikulecký, P., Bodnárová, A., Olševičová, K., Ponce, D., Haviger, J., 2008. Application of multi-agent systems and ambient intelligence approaches in water management. 13th IWRA World Water Congress, Montpellier (France).

Nichita, C., Oprea, M., 2007. Water Pollution Diagnosis with a Multi-Agent Approach. In Proceedings, Artificial Intelligence and Soft Computing.