Izquierdo Sebastián, J.; Montalvo Arango, I.; Pérez García, R.; Ayala Cabrera, D. (2015). Multi-agent simulation of hydraulic transient equations in pressurized systems. Journal of Computing in Civil Engineering. 04015071:1-14. doi:10.1061/(ASCE)CP.1943-5487.0000549.

Multi-agent simulation of hydraulic transient equations in pressurized systems

Joaquín Izquierdo[1], *Idel Montalvo*[2], *Rafael Pérez-García*[3], *David Ayala-Cabrera*[4]

[1] *Professor, FluIng-IMM, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia (Spain) (corresponding author). E-mail: jizquier@upv.es*

[2] Ph.D., Ingeniousware GmbH, Bahnhofstraße 4a, 76137 Karlsruhe, Germany. E-mail: imontalvo@ingeniousware.net

[3] Professor, FluIng-IMM, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia (Spain). E-mail: rperez@upv.es

[4] Ph.D. Candidate, FluIng-IMM, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia (Spain). E-mail: daaycab@upv.es

Abstract

Computational modeling pervades virtually every industrial process. By using numerical representations of the behavior of elements that constitute a system it is possible to obtain efficient and safe designs. Moreover, system operation can be better defined by using such models, thus enabling greater reliability and control. In this paper, the use of agents to solve the equations describing fast transients in water networks is investigated. As the simulation of hydraulic transients in pressurized systems is a naturally distributed problem, the authors argue that a multi-agent based system is very suitable for the solution of this complex engineering phenomenon. A hybrid solution is built by deploying agents to work with sets of equations describing hydraulic transient behavior in pipeline systems. The details necessary to assemble a complete and lubricated machine to model the complex phenomenon of hydraulic transients in

pressurized systems are described. As a result, this research develop a platform that constitutes an efficient and versatile tool of great interest for water supply managers when analyzing water hammer effects in their networks.

INTRODUCTION

Water distribution systems (WDSs) are complex distributed infrastructures. The network of a water distribution system is constituted by an interconnected and intricate set of pipes that includes elements with complex behaviors (Izquierdo et al. 2012; Yazdani and Jeffrey 2011). Since the main purpose of WDSs is to provide the public with a service of first necessity, the strategic, social, environmental, and sanitary importance is clear.

Managers of water supply companies are concerned about the lack of integrity of their systems for a number of reasons. The consequences of failure are manifold. Service irregularities, sometimes acceptable, may result in severe service disruptions. Pipe breakages may cause large investment losses, water wastage, and major third party damages. Cracks in pipes may cause two effects of great concern in urban water management, namely, water leaks, which represent a substantial, continuous, and imperceptible waste of water; and pathogen intrusion, which impairs water quality and represents a serious risk to human health.

Despite its importance, transient behavior, especially fast transients (also known as water hammer) still represents a challenge for many water companies. Powerful tools for making decisions about this problem are necessary.

Today there is generalized unanimity about the need for computational aid to cope with the overall complexity of the phenomenon. However, work on calculations to build an efficient hydraulic transient simulator is still under study (Guidaoui et al. 2005).

To the authors' knowledge, none of the existing software tools in the market has addressed the problem from a perspective that fully takes advantage of the distributed nature of the problem by using an agent-based approach. The objective of this paper, which is a thorough extension of (Izquierdo et al. 2010), after having completed the project therein presented, is to make known a hydraulic transient analysis tool based on multi-agent concepts.

In the next section, by presenting clear parallelisms between hydraulic transients and the multi-agent philosophy, the authors motivate the approach. Afterwards, they concisely describe the concept of a multi-agent based system; this section is included for the sake of completeness, since the advances in a multi-agent based systems presented in this paper are primarily with respect to the representation of engineering knowledge related to hydraulics. Then a new section provides a dictionary to translate hydraulic transient elements to multi-agent language. This enables the authors to provide in another section the necessary details that facilitate the definition of a multi-agent platform to simulate hydraulic transients in pressurized systems. Some notes stating the validity of the approach are also given. The last section offers conclusions and closes the paper.

EMBEDDING HYDRAULIC TRANSIENTS IN MULTI-AGENT PHILOSOPHY

Analyses of most hydraulic transients in pressurized systems are carried out assuming one-dimensional flow and are based on the continuity and momentum equations describing the general behavior of a fluid within a (cylindrical) pipe in terms of two dependent variables, namely, $H(t,x)$, piezometric head, and $V(t,x)$ fluid velocity (Abreu et al. 1995; Chaudhry 1986; Wylie and Streeter 1993).

The continuity and momentum equations applied to the pipe constitute a system of first order partial differential equations that can be written (Izquierdo et al. 2004) using matrix notation, such as

$$\begin{pmatrix} H \\ V \end{pmatrix}_t + A(V)\begin{pmatrix} H \\ V \end{pmatrix}_x = B(V) \qquad (1)$$

where the sub-indexes *t* and *x* denote partial derivatives with respect time and space, respectively, and

$$A(V) = \begin{pmatrix} V & a^2/g \\ g & V \end{pmatrix}, B(V) = \begin{pmatrix} -V\sin\theta \\ -fV|V|/(2D) \end{pmatrix}. \qquad (2)$$

Celerity or wave propagation velocity, *a*, friction, *f*, diameter, *D*, and slope, θ, are parameters of the pipe that are constant in time. These parameters, however, may be different for different pipes. Finally, *g* is the acceleration of gravity.

Expression (1) is a non-linear, hyperbolic, partial differential equation (PDE) system, since matrix $A(V)$ has real simple eigenvalues for each V. In effect, the characteristic equation of $A(V)$ is

$$(V - \lambda)^2 - a^2 = 0, \qquad (3)$$

with roots $\lambda = V \pm a$, which are real and different, taking into account that a is at least one order of magnitude larger than V in pressurized systems.

In most practical cases $V \ll a$, and as a result, the acceleration convective terms are negligible. Moreover, the slope term can be neglected, and it is common practice in hydraulic engineering to use flowrate Q instead of V as the flow variable. Finally, assuming incompressible flow, and a pipe constant cross section, A, equation (1) can be written for a pipe as

$$H_t + \frac{a^2}{gA} Q_x = 0 \quad . \quad (4)$$
$$Q_t + gAH_x + \frac{fQ|Q|}{2DA} = 0$$

These (wave) equations govern the behavior of perturbations (travelling waves) through the water-pipe system. Nevertheless, other elements such as junctions, feed points, pumps, and valves located at the joints (and thus at the ends of the pipes) are also important parts of a hydraulic network. The behaviors of these elements are referred to as boundary conditions. From a topological point of view, boundary conditions may be classified (Spellman 2013, Izquierdo and Iglesias 2002) as parallel or series elements. Typical constitutive equations for parallel elements are of the form $H(\mathrm{P}) = \mathbf{F}(Q_\mathrm{e})$, where Q_e is the flow through the element, and H is the piezometric head at point P. For series elements, the constitutive equations take the form $\Delta H_\mathrm{e} = H(\mathrm{P}_1) - H(\mathrm{P}_2) = \mathbf{F}(Q_\mathrm{e})$, where ΔH_e is the pressure gradient through the element located between points P_1 and P_2, and Q_e is the flow through the element. In both cases, $\mathbf{F}(.)$ describes the hydraulic behavior of these elements and ranges from simple relations to systems of (mixed) algebraic and/or differential equations.

Until this point, this paper has used a register that is familiar to engineers and somewhat familiar to mathematicians, but is distant from the terminology used by many experts in multi-agent based systems. In the following paragraphs the paper will mix together engineering and multi-agent terms through a joint vocabulary that will eventually be developed into fully multi-agent-based phrasing. The main idea of this paper is that hydraulic transients can be addressed using multi-agent techniques, and that the advantages are manifold.

Since von Riemann (von Riemann 1869) obtained the solution of the wave equation,

$$u_{tt} - a^2 u_{xx} = (u_t + au_x)\,(u_t - au_x) = 0, \quad (5)$$

by overlapping the solutions carried by two travelling waves,

$$u(x,t) = F(x + at) + f(x - at), \qquad (6)$$

it is well-known in transient phenomena, and in hydraulic transients in particular, that the transmission of perturbations is a phenomenon of the transmission of information – that is to say, a communication phenomenon. Given the information corresponding to an initial condition, a given state, it is possible to build its future evolution through the so-called characteristic lines (characteristic hyper-surfaces, in general) by simply transmitting information.

The representation can be made still clearer by examining the most utilized numerical method for the resolution of the wave equation, the method of characteristics (MOC). In the MOC, space and time are discretized (see afterwards for details).

This discretization generates a discrete group of calculation points in a given pipe that are active elements, owners, and carriers of information that changes to defined impulses with time. The time discretization sets the schedule of the combined activity. The emergent behavior of the calculation nodes of a pipe enables the pipe to be considered as an agent of another level whose behavior still needs new rules to communicate with the remaining elements of the network, namely, hydraulic devices located at the ends of the various pipes integrating the network.

These new breeds of agents are called boundary conditions (BCs). They have specific behaviors, not necessarily autonomous, and interact in the joint environment, partly in a predefined way, in accordance with appropriate design considerations. Moreover, the appearance in the network of loading conditions far from the design point (due to abrupt changes in demand, fire events, maneuvers, etc.) means that the behavior of such elements is also conditioned by certain successions of events.

Thus, the elements that are involved possess individual behavioral rules that can be influenced or modified by the behavior of other elements. Moreover, a fundamental part of the phenomenon is the succession along time of queries and answers, that is to say, an exchange of information among the different elements of the system. This permanent dialogue among such elements characterizes the events that take place. Information among the elements is transmitted on request. The obtained information enables the various elements to complete their own information, define their behavior, and produce answers that, in turn, feed other elements.

By the end of this section the paper has reached the point of exclusively using multi-agent terms. In the next section, the main multi-agent concepts are presented.

MULTI-AGENT BASED (MAB) SYSTEMS

Computer programs have played an important role in the study of complex systems. However, the actual process of writing software is a complicated technical task with much room for error. The multi-agent philosophy adopts a modeling formalism based on a collection of independent agents interacting through discrete events (Stone and Veloso 2000; Weiss 1999; Wooldridge 2002). Simulation of discrete interactions between agents thus perfectly fits the engineering almost-universal process of discretization used to solve most problems defined by systems of coupled differential equations.

In the following paragraphs concise ideas about the main ingredients of a MAB system are provided. To try to establish early connections with this context several examples are mentioned. Most of this information is expanded in the next section.

One definition of agent. An agent is any actor in a system, any entity that can generate events that affect other agents in the system, including itself. In the MAB architecture considered 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, synthesizes the results, and optimizes the solution through coherence and coordination. Semi-autonomous refers to the fact that some agents have uncertain knowledge about the environment, so that they can ask the human operator for specific action to take into account current or future scenarios (Cohn et al. 2011). For example, in the context of this paper context, some devices (such as valves and pumps) may need input from the operator of the system in specific circumstances.

Types of agents according to their complexity. Agents define the basic objects in the system – the simulated components. Agents may be simple or compound. In this latter case, agent behavior is defined by the emergent actions of the agents they contain. The simulation occurs in the modeled world itself, and it is frequent to speak of agents as living in an environment, which, in its turn, can be an agent itself. The whole system is an agent following specific scheduled actions.

Breeds or categories of agents. Agents belong to different 'breeds', categories, or species. Agents from different breeds behave differently. In the problem considered here, agents are pipe discretization points, consumption nodes, connecting pipes, supply sources, various devices, ground patches containing the network – as well as district metered areas, which are set of nodes, pipes, sources, and patches.

Creation of agents. Agents are created and incorporated in the platform in an individual or collective manner. Some agents are created using the built-in tools that enable locating specific elements at specific coordinates in a geo-referenced system. Other agents are created in an automatic way when certain processes are initialized or triggered. For example, pipe calculation nodes are created when a suitable discretization is defined. As another example, various graphs or outputs of data (other breeds of agents, as noted later) are created at the user's request, which in turn, can interact with their properties, etc.

Properties of agents. A newly created 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. These agent properties can be individual, used by the agent in an exclusive way, or collective. Properties can be defined for many agents simultaneously. Properties are used by agents in their relationship with other agents, and encapsulate the protocols of information exchange. As a result, during the process, each agent can recognize an approaching agent; dialogue appropriately with that agent; and offer the required answer.

Decision rules. Most MAB applications deal with very simple agent models, mainly expressed in terms of simple behavior and decision rules. The degree of sophistication of the agent model depends on the scale of the simulation and the complexity of the problem. In the case of hydraulic transients, agent models must be based on mathematical models (usually in terms of systems of algebraic and differential equations) and on rules that range from simple (describing plain autonomous actions) to sophisticated behavior (involving the effects of other agent activities). As stated by Wooldridge (2002) some agents are passive, others are reactive and, finally, some are proactive. However, independently of their specific characteristics they all have important common characteristics, i.e., they are all situated in a geo-referenced space, they must be somehow aware of other agents in this space, and they interact within the environment. It is thought-provoking to note here that this classification perfectly suits the classical boundary condition classification used for years in hydraulic engineering and other fields in nonautonomous, autonomous, and dynamic boundary conditions (Abreu et al. 1995; Evangelisti 1969).

Figure 1. Example of agent behavior

Behavior of an agent. An agent is also associated with a specific behavior. Behaviors can be represented by sets of multi-layered directed graphs. Each graph is made from a set of objectives for the agent to reach (see a general illustration in Figure 1). Behavioral graphs are data structures used to define complex agent behaviors. These graphs are made from nodes representing objectives (see various general objectives in Figure 1). Objectives are structured in a hierarchical manner such that elementary objectives are associated with actions the agent can execute. Different objectives are connected by links that may be simple (thin arrows) or multiplexers (thick arrows), depending on the number of transmitted signals. In general, agents own sets of objectives that can be either simple or compound according to their needs.

Objectives are also associated with rules describing the activation, execution, and completion of the objective. Activation rules, closely associated with 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.

The way rules apply depends on the time, agent's state, and environmental state, and in the architecture herein described, the user can also introduce modifications online (semi-autonomous agents). An agent selects its current objective (according to the graph structure) with respect to previously executed objectives, and with respect to priorities regarding the abovementioned rules.

As can be observed in Figure 1, the structure of the multi-layered graph may contain behaviors at different levels of abstraction. Compound objectives can be devised as decomposable structures representing sub-behaviors, with similar structures. Several specific examples are provided in the next section to illustrate the way these behaviors are implemented.

Schedule of events. 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 a specific time, and advance alongside events scheduled at successive times. A schedule is another data structure that combines actions to be executed in a specific order. The passage of time is modeled by the execution of the events in a sequence. Instructions are given to hundreds or thousands of independently operating agents. This makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from the interaction of many individuals.

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

The aim of this section is to briefly present the most elemental concepts of MAB systems, since the main objective is the representation of engineering knowledge related to hydraulics. Thus, this paper will not develop this further. Instead, in the following section a number of clues and details aiming to help an audience with knowledge of MAB obtain an insight into the engineering implementation of the approach described are provided. Before describing the specific architecture developed by the authors to simulate hydraulic transients in water networks, it is worth emphasizing the growing importance of MAB systems in various applications.

The growing trend in recent years (Dibley et al. 2011, Nguyen et al. 2012, Ruiz et al. 2014, Zhang et al. 2014), is to include multi-agent techniques as an interesting alternative for solving complex problems. Multi-agent techniques have been widely used in water the field (the authors' field of expertise) such as allocation of scarce water (Hailu and Thoyer 2005); water and waste water control system architecture (Maturana et al. 2006); control systems for municipal water (Kotina et al. 2006); water pollution diagnosis (Nichita and Oprea 2007); optimization of water networks (Cao et al. 2007); water management at river basin scale (Mikulecký et al. 2008); river basin water allocation management (Yang et al. 2010); water availability (van Oel et al., 2010); inter-basin water transfer (Huang et al. 2011); monitoring irrigation systems (Zhao et al. 2011); division of water supply networks into district metered areas (Herrera et al. 2011, 2012; Izquierdo et al. 2011); water rights transfer market issues (Igual-Herrero 2012); identification of buried assets in water distribution systems (Ayala et al. 2011, 2013); and design of water supply networks (Montalvo et al. 2011, 2014).

MAB PLATFORM FOR HYDRAULIC TRANSIENT SYSTEM

The development of a MAB platform for hydraulic transient simulation is a challenge because two sets of very distinct concepts that neither share the same problems nor the same concerns, must be put to work in a synergy that has to conciliate these differences and enable the fruitful interactions sought. However, there is a clear meeting point, namely, the distributed nature of the problem.

From the software perspective, a MAB system is the natural framework to implement parallelization. It means a significant advantage for hydraulic transient analysis because of the 'gained' calculation time. Nowadays, mathematical models of water networks are moving from an off-line perspective to an on-line context where they are almost required to give responses in real-time. For relatively complex networks the response time could be excessive considering the requirements. However, parallel calculation can bridge the gap. In a 'pipe agent' – described later – for example, the transient analysis in all its discretization points (another breed of agents described in the next section) can be executed independently from any other agent, provided its two (upstream and downstream) boundary conditions are known. Thus, for a specific point in time, calculation could be performed in a 'pipe agent' without waiting for other 'pipe agents'. The same principle applies for running parallel calculation at 'consumption node agents'. The global environment just needs to consider that all the agents should finish their calculations one time step before starting calculations for the next time step.

In the following paragraphs this paper provides the necessary details to accommodate the problem tackled here with the proposed methodology for handling the problem, which has been concisely described in the previous section. The objective is to provide the relevant connections between the ingredients of a MAB system and the engineering concepts related to hydraulic transients.

The following subsections describe the various agents that such a platform has to accommodate with respect to hydraulic specifications and implementations.

The most elemental agents – calculation points

The MOC is the most popular method to numerically solve the set of equations (4). For the sake of simplicity, this research will use the simplest scheme for the numerical solution of (4) using the MOC. Using this method, space (pipe of length *L*, represented by the base line between 0 and *L*, in Figure 2) and time (in an interval [0, *T*], for any arbitrary value *T*) are discretized (grid structure of the problem domain in Figure 2). The numerical solution is calculated on the grid vertices. The Courant-Friedrics-Lewy (CFL) condition – necessary for the stability of the scheme presented here – relates space and time discretizations, Δx and Δt, respectively, making the characteristic lines, C+ and C- with slopes $+a$ and $-a$, respectively, a being the wave speed. Specifically, the CFL condition states that $\Delta t \leq \Delta x/a$. This generates a discrete group of agents (calculation points) in a given pipe (thick points on the base line in Figure 2), which, with the discrete passing of time, are represented by the corresponding points on their verticals on the consecutive horizontal lines in Figure 2.

Figure 2. MOC discretization, characteristics, calculation points and boundary conditions

The built-in behavior of such elements and the mechanism of information transfer are very well-known. The following concise description is included here. Let us first consider the case of a typical inner element, P, representing an agent at a given time step (see point P on the fourth line in Figure 2). Agent P receives information, through the characteristic lines C+ and C-, from its neighbors, X and Y, (referring to the previous time step, located on the third line for the current example) and uses this information to make up and adopt its current state. Specifically, in the case considered here, the state of one of those (inner) elements P is defined by its state values $Q(\mathrm{P})$ and $H(\mathrm{P})$ obtained by solving the linear system

$$\begin{aligned} Q(\mathrm{P}) + C_a H(\mathrm{P}) &= C_p \\ Q(\mathrm{P}) - C_a H(\mathrm{P}) &= C_n \end{aligned}, \qquad (7)$$

where

$$\begin{aligned} C_p &:= Q(\mathrm{X}) + C_a H(\mathrm{X}) - RQ(\mathrm{X})\,|Q(\mathrm{X})|\,\Delta t \\ C_n &:= Q(\mathrm{Y}) - C_a H(\mathrm{Y}) - RQ(\mathrm{Y})\,|Q(\mathrm{Y})|\,\Delta t \end{aligned}. \qquad (8)$$

These values are obtained by using the rectangle rule (a first order integration rule) to integrate the ordinary differential equations that apply over the characteristics (Abreu et al. 1995; Chaudhry 1986; Wylie and Streeter 1993).

The constants are $C_a = gA/a$ and $R = f/(2DA)$; Δt is the time step.

The elementary algebraic system (7) defines the behavior of a typical inner node, P, of a pipe in the sense that the system allows P to obtain values for its associated variables, namely the flow Q(P) and the piezometric head H(P), using similar information transmitted from X and Y through the characteristic lines.

In this way, the element is updated and can transmit this up-to-date information following requests from other elements. These simple actions can be easily rendered (see Figure 3) into the template of the behavioral directed graph given in Figure 1.

Figure 3. Behavioral directed graph for a calculation point

Let us now consider the end points of a pipe, such as the points noted by P on the vertical lines over 0 and L in Figure 2. These points only receive information from one (inner) neighbor and need some (outer) additional information to define their behavior. This is addressed in the next two subsections.

Importantly, the discretization provided by the MOC establishes, through the time step Δt, the schedule for the discrete events that will occur over time. Other agents with behaviors depending on time will synchronize their clocks with this general schedule. This aspect is considered later, within the appropriate context, in the subsection after the next.

Pipes as higher level agents

The emergent behavior of the inner nodes transforms a pipe into a higher level agent whose entire behavior needs new rules to communicate with the rest of the elements in the network. Those other elements from the network to which the pipes are connected are known by the general name of boundary conditions (solid end points at 0 and L in Figure 2 at $t = 0$, and the respective points on their verticals for other time steps). Their behaviors are mutually influenced. As a consequence, it is essential to define not only all the potential elements but also their possible interrelations.

The behavior of a pipe can be described very simply, since it communicates with other (target) agents exclusively through points 0 and L. This communication uses the two characteristic lines that leave outwards from these points, specifically a negative characteristic outwards from 0, given by the second equation of (7), and a positive characteristic outwards from L, given by the first equation of (7). These characteristic lines provide the target elements with specific information, a relation between flowrate and piezometric head that the target elements have to accommodate with their own information. In the following paragraphs the paper introduces these target elements and describes how they interrelate with pipes.

Consumption nodes, higher-level agents that generalize calculation nodes

Pipe junctions, which may have some associated consumption, are the most common boundary conditions in a water network. The activity of a consumption node typically consists in the negotiation of the characteristics of the various pipes joining at the junction, the preparation of the information that determines its behavior, and the dissemination of specific information to the connected pipes in an appropriate way.

Let us consider a consumption node P, with associated demand $Q_P(t)$, a known function of time, connecting N pipes. Its behavior uses the characteristic equations of these pipes, as explained previously. For each of the N_{in} incoming pipes to P, a positive characteristic (similar to the first equation of (7)), and for each of the N_{out} outgoing pipes from P, a negative characteristic (as in the second equation of (7)) are considered. Note that $N = N_{\text{in}} + N_{\text{out}}$. In addition (neglecting head losses at the node, something typically accepted on engineering grounds, especially when calculations are performed with suitable safety margins) all the values of $H(\text{P})$ for the N pipes coincide with the piezometric head, H, at the node where they all meet. Finally, the continuity equation for the node, stating that the net (positive for incoming and negative for outgoing, or vice versa as initially stated) amount of flowrate equals the demand $Q(t)$ is considered. Using all this information, the head H at the node may be obtained from

$$Q_{\text{P}} + C_A H = C_+ - C_-, \tag{9}$$

where C_+, C_- and C_A account for all the coefficients C_p, C_n and C_a of the pipes meeting at node P, with instant demand Q_P; see a detailed description in (Izquierdo and Iglesias 2002, 2004) and in (Abreu et al. 1995).

Finally, using the characteristic lines of all the pipes, the flowrate delivered to or taken out from the demand node by each pipe can be calculated. This completely defines the state of the demand node.

Neither from the conceptual, programming, nor the physical points of view are there substantial differences regarding the behavior of the inner nodes of a pipe, except for the fact that a consumption node can have, as suggested by its name, an associated consumption, and more than two pipes may join in a consumption node (that is to say, it may have to handle more than two characteristic lines).

Various devices, the specialist agents

In contrast to consumption nodes, other boundary conditions are more complex. Various devices and combinations of them produce, damp, amplify, and control perturbations – and so generate in hydraulic installations conditions outside their design regimes.

The individual devices (pumps, valves, air vessels, etc.) are described in a satisfactory way in the literature using different types of models, in general, steady state models or lump models (Abreu et al. 1995; Izquierdo and Iglesias 2002; Thorley 1991). Most of these models are described by sets of mixed algebraic and differential equations. If a function of time is used to (partially) describe the behavior of one of these devices, then this function must accommodate the general schedule established by the MOC. Since many of these time functions are given in a discrete form, suitable interpolation techniques are required. If the solution techniques involve derivatives, differentiable forms are advisable. As an example, the behavior of a pumping station may be modeled by the following set of equations.

- The energy equation requires that the difference between the discharge head, H_d, and the suction head, H_s, equals the difference between the dynamic head of the pump, ΔH_p, and the total head loss, ΔH_e, at the resistive components of the element:

$$H_d - H_s = \Delta H_p - \Delta H_e . \qquad (10)$$

- The head loss across the element can be characterized by

$$\Delta H_e = K_e Q|Q| , \qquad (11)$$

where K_e = joint head loss coefficient of the resistive components of the element.

- The dynamic head of the (equivalent) pump can be represented (Chaudhry 1987; Wylie and Streeter 1993) by

$$\Delta H_p = H_r h = H_r\left(\alpha^2 + q^2\right) f(\theta), \qquad (12)$$

where H_r = rated pump head, α = relative pump speed, defined by $\alpha = \omega/\omega_r$, being ω = pump rotational speed and ω_r = rated pump rotational speed, q = ratio between Q and Q_r = rated flowrate, $\theta = \mathrm{atan2}(\alpha/q)$, and $f(\theta)$ the dimensionless head of Suter curves (Marchal et al. 1965).

- The torque equation, $M = -I(\mathrm{d}\omega/\mathrm{d}t)$, after integration using a second-order trapezoidal approximation, and using the dimensionless Suter curve for the torque, $\varphi(\theta)$, may be written as

$$\lambda\alpha + \left(\alpha^2 + q^2\right)\varphi(\theta) + V = 0, \qquad (13)$$

where $\lambda = (2I\omega_\mathrm{r})/(M_\mathrm{r}\Delta t)$, being I = inertia of the impeller, entrained fluid and rotating parts of the pump, M_r = rated torque, Δt = time step used by the MOC, and V a constant that appears when performing the integration depending on λ , α and M_r.

This set of equations is clearly non-linear. Solutions that use, for example, the Newton method, need a differentiable representation of the Suter curves. As these curves are given by discrete points, a suitable technique, such as cubic splines for example, must be employed to obtain differentiable expressions for the curves.

However, the main difficulty derives from the difficulty in programming the compound boundary conditions that appear in actual facilities. Programming each possible combination of elements will produce many different routines with many common lines of code, and this makes programming inefficient and obsolete. Maintaining simple codes for the elements and linking them with short pipe sections necessarily leads to the so-called curse of the short pipe, a consequence of the CFL condition, which is necessary to guarantee stability and which takes the calculations to inadmissible situations in terms of computational resources (time and memory). The above mentioned problems in the definition of compound boundary conditions are obviated with the MAB approach proposed here as an elegant and efficient approach. Simple elements are defined in an appropriate way that is, at the same time, code-unique. That is to say, there is no code repetition. And, by virtue of the multi-agent approach employed, the combinations of different simple elements in one location are carried out through the introduction of new agents, called facilitators, who harmonize the traffic of questions and answers among the simple elements. In other words, the facilitators manage the dialogue.

The facilitator, a broker agent

A facilitator is a new class (breed) of agent designed to put various simple devices in contact to 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 modifies the internal variables necessary to carry out its specific function in an autonomous way. It knows the elements it has to put in contact, what to ask any of the elements, how to prepare the necessary information, how to negotiate with them, and how to respond to each with the requested information.

Figure 4. Facilitator behavior

The behavioral graph of the facilitator can be seen in Figure 4. As with other agents, this graph consists of nodes representing objectives. The hierarchical structure of the objectives includes the elementary objectives associated with actions the agent can execute. In this case, the elementary objectives are: ‘Schedule_updating’; ‘State_updating’; ‘Third_Party_Idenfication’; and ‘Third_Party_Query_Retrieval’. ‘Schedule updating’ simply synchronizes with the global schedule previously defined. ‘State_updating’ recalculates all the variables defining the agent state. ‘Third_Party_Identification’ recognizes the type of agents (devices) that the facilitator has to put in contact. ‘Third_Party_Query_Retrieval’ asks those agents for the necessary information for the facilitator to prepare the joint response. In the case of the last two objectives, multiplexers are used since various signals make up the flow of information. Finally, some feedback may be established to fine-tune the facilitator’s joint proposal to meet all the requirements as much as possible.

As said, the structure of the multi-layered graph may contain behavior at different levels of abstraction, the compound objectives are devised as decomposable structures representing sub-behaviors. In Figure 5 a graph showing the facilitator’s sub-behavior ‘Pundit’ is shown, where ‘Mediation_Preparation’, ‘Response_Elaboration’ and ‘Joint_ Proposal’ are the objectives to meet. The ‘Mediation_Preparation’ objective consists in univocally identifying the agents that the facilitator has to put in contact. The resources for this objective consist of a database of possible candidates (including elements such as pipe, reservoir, pump, valve, and air vessel); the identification of the intervening agents according to their credentials corresponds to the execution rules for this objective; finally, the completion rules prepare the necessary details to start with the next objective. To meet the ‘Response_Elaboration’ a new database with the necessary variables for the agents involved constitutes the set of resources; the execution rules then suitably combine those variables by performing the necessary calculations; finally, the complete set of data for the next step is prepared following the completion rules. The final objective for this subtask, ‘Joint_Proposal’ delves into a set of predefined protocols, constituting the resources of this objective; and the execution rules then prepare these protocols and associate in an orderly manner the obtained values for the pertinent variables; the completion rules formalize the way in which the sub-task passes the mediation proposal to the next step. The transcription of the numerous lines of code or pseudo-code that embody all these aspects will not be detail here.

Figure 5. Sub-behavior of the facilitator

As stated previously, facilitators manage to overcome two important problems affecting similar tools found in the market, namely, repetition of code and use of short pipes. The former makes packages obsolete and inefficient in terms of code writing and debugging, and, above all, in terms of extensibility. The latter, has two other pernicious effects. Computer codes using short pipes have two main problems when trying to overcome the CFL condition. Either they simply cannot perform certain hydraulic simulations due to a lack of computational resources or, what is worse, they produce unrealistic results by performing artificial adjustments that are opaque to 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 less justifiable adjustments (such as assigning minimum lengths to pipes that are so short that pose problems for CFL condition compliance).

The observers

Within this category various breeds of agents whose main objective is to facilitate communication between the user and the environment where the operating agents live may be considered. These agents range from simple error flags and plain text boxes to display messages, to simple and not so simple graphs, charts, and automatic reports fully customizable by the user.

The paper does not develop further on this category. Various detailed descriptions, outside the scope of this paper, may be found elsewhere.

The user

As said, the user is surely the most important agent in this MAB system. Many tasks are user-dependent.

- The user can create, customize, and define many aspects, trends, agents, observers, etc., and may even set the pace of the schedule of the simulation by suitably manipulating the CFL condition.
- The user can group agents to form a new higher level agent that can perform joint actions on all the agents belonging to a certain group. For example, the friction factor or the roughness of all the pipes belonging to a certain group of pipes, or a whole sector or a district metered area, may be changed with just one user action if new information regarding the state, age, etc. of the pipes in a certain area changes after a new study.
- The user can also establish scenarios for easy comparison, and make decisions about various aspects such as the optimal protection strategy for the network.
- The user can create and maintain suitable databases in a GIS format that are connected with graphical representations.

In summary, in the problem considered here, agents are the calculation points within a pipe: the consumption nodes, connecting pipes, supply sources, devices (reservoirs, tanks, valves, pumps, air vessels, surge tanks, one-way surge tanks, etc.) and ground patches containing the network; as well as the district metered areas (DMAs) which are sets of nodes, pipes, sources, and patches; and, last but not least, the facilitators. The entire network is itself an agent following specific scheduled actions. In the case of pipes, DMAs, and the entire network, behaviors are defined by the emergent actions of the agents they contain. The user may also be an active (and certainly the most important) agent.

Finally, the platform (described in full in the next section) is the world or environment in which live the agents that participate in the simulation of hydraulic transients in complex systems.

THE PLATFORM AND ITS VALIDATION

In this section the authors first succinctly describe the platform where the presented ideas have been implemented, and explain how the platform is used in one of the various real-world case studies the authors have performed in recent years (see right panel in Figure 6).

To complete this section these researchers develop a second subsection that provides a number of additional details that validate the developed applications and show that conventional software can be constructed out of agents, and software engineering can be used in this endeavor (Hunhs et al. 2003).

Figure 6. Platform IDE

The IDE

Figure 6 presents a global view of the IDE platform (integrated development environment). The main components are briefly described below. Only the most relevant ones are mentioned here.

* Various elements – most being typical in Windows applications – integrate the main menu. In Figure 6, the ‘Edit’ toolbar (the ‘Edit’ tab is highlighted) includes the ‘Visualization’ (leftmost elements, starting by ‘Normal state’ and including the ‘Insert profile’ icon) and the ‘Network Component’ (rightmost elements, from ‘Demand node’ to ‘Delete’) tools. The ‘Analysis’ toolbar (presented later) enables control of the main parameters of the simulation.

* A project is currently opened and occupies the main screen area (two panels to the center and the main panel to the right). The main panel contains the network. Drag-and-drop facilities enable on-screen construction or deletion of a network. Alternatively, nodes, pipes, and boundary devices together with their data can be directly imported from EPANET (Rossman 2002). In Figure 6 the considered network has already been loaded and there are three tabs in the right panel:
  - **Mapview** tab: presents a plant view of the network (it is the active one in Figure 6); observe that one of the pipes has been selected and its properties are displayed in the 'Properties' control located on the left;
  - **Tableview** tab: shows information in tabular form about the elements of the activated layer in the 'Layers Control' window;
  - **Analyzer** tab: enables the main simulation functions, including the play and stop buttons, located for convenience just to the right of the tab together with the current simulation time.

* Profile #*n* is a type of view that presents a detailed profile (a connected path) of the network being analyzed; an arbitrary number of profiles may be defined and monitored. Two profiles, stacked vertically, are shown at the center of Figure 6. A second line parallel to a profile denotes the cavitation (vapor pressure) line. Piezometric grade lines should never touch this line (in a well-designed installation).

* Project Manager: enables to manage all the defined and active elements in the environment.

* System controls: windows (to the left of Figure 6) that show specific system information:
  - Layers Control: to activate the various layers defining the analyzed system as in other GIS environments.

- Properties Control: to show the properties of the selected network element.
- Color Scale Control: to select color scales for graphical representation.

* Communication tab: allows further interaction with the user (orange rectangle located on the bottom left in Figure 6).

These elements are in constant interaction and constitute the architecture of the platform. Nonetheless, on a more global level, the platform can also be seen as two distinct parts; first, the framework for the definition of the various agents and their behavior; and, second, the set of tools to facilitate user manipulations. These tools are spread in various primary and contextual menus that simplify their use as much as possible.

Once agents have been defined, the ‘Analysis’ menu option associated with the main document deploys the ‘Analysis’ toolbar shown in Figure 7. A number of characteristics related to the multi-agent system schedule are shown/selected here.

This toolbar shows the initial and final time of the simulation and enables the user to: define a maximum calculation time (duration); toggle to monitoring activities; specify waiting time; select the specific formula for calculating head losses; select the algorithm to be used and type of regime (the associated transient being the main objective); and introduce specific intervals of calculations for hydraulic and water quality variables. Once these parameters have been established, the ‘play’ button, located on the ‘Mapview’ panel, triggers the simulation.

Figure 7. Available analysis options

The user then takes his or her role as an agent and starts observing the graphical representation of the simulated transient. This representation, as observed in Figure 8, is twofold. This figure shows a snapshot of the simulation with a view of the two profiles previously defined, and the ‘Mapview’ of the network. Observe the hydraulic grade lines for the profiles (including so-far maximum and minimum envelopes), and the colors of the nodes according to the color codes used in the ‘Color Scale’ control (various node colors in Figure 8 correspond to values defined in the respective control and indicate various head levels). Both the profiles (with their moving piezometric lines) and the ‘Mapview’ (with the changing color of the nodes) convey a deep insight into what is happening. In effect, during runtime, the moving piezometric lines in the profile windows and the changing colors of the nodes present a qualitative dynamic movie of what is really happening. Unacceptable situations may be clearly identified from piezometric lines going, for example, below the pipe line (or even below the cavitation line), and from the colors of the nodes changing according to the palette defined in the ‘Color Scale’ control. According to the evolution and performance of the system, the user may decide between two options: stop the simulation, if anything undesirable is happening, then take control of the system and introduce the necessary modifications; if the system works properly, leave it to evolve until transient completion – and then make a detailed examination of the results. The ‘View’ menu option of the main document enables this task. With this option the user can observe various kinds of graphs and charts of time histories of the variables associated with the devices previously selected for monitoring.

Figure 8. Snapshot of platform IDE showing grade lines on the profiles and color-coded points regarding the network mapview

The case study

The case study is a small network fed by a tank. This network is a district metered area (DMA), one of the sectors of a larger WDS. Two control cut-off valves located to the right and the bottom of the panel, which enable the isolation of the considered DMA, are responsible for the transients generated on closure. Due to space limitations, the authors spare here the reader all the specific data of the elements of the DMA (node elevation and demand; and length and diameter of the pipes, being the most important).

The managers of the network were interested in simultaneous optimal closure of both valves. Since isolation of the sector could be carried out at any time, the maneuver effects strongly depend on the current demand associated to the current operation point on the demand curves. In this case the demand pattern for all the nodes was the same, the base demand varying among the various nodes between 5 and 20 l/s.

The constraints for the study implied three different main aspects: 1) the pressure should not exceed 100wcm anywhere in the network; 2) the pressure should not go down 10wcm in any of the nodes; 3) the closure time should be minimized since the maneuver could be the action in response to a vulnerability emergency.

To help show how the platform can be used, the authors present now a simplified version of the study. It is supposed that the network and its elements' characteristics have already been introduced. See again Figure 6 where, in addition, two profiles have been selected.

To develop the project, first various scenarios must be considered based on the current consumption in the network. Among those scenarios the more severe should be considered to develop the closure strategy. In principle, all the less severe scenarios would be covered with the proposed maneuver (something that must anyway be checked with the corresponding simulations). In the present study the most severe scenario corresponds to the moment of highest consumption, since the bulk of the water running into the network to be controlled (eventually stopped after closure) is larger. The steady state for this scenario has already been calculated. The next step is to consider the valve characteristics. In this case, both valves were identical and exhibited the typical low regulation capacity of many valves in the market. As a result, even long uniform (linear) closure times produce severe transients, since the effective closure only develops during the last part of the closure, most of the early closure time being completely inefficient. See Figure 9 representing the characteristic curve of the valves and the effect on the flowrate through one of them along the closure.

Figure 9. Characteristic of the valves and effect on the flowrate

Simulations for a linear closure maintaining the pressure within the specified limits produced closure times (beyond 120s) that were not acceptable from the emergency point of view. The final proposal overcoming this problem consisted in a two stage closure in which the valves would close from fully open to 15% in the first 3 seconds, and then close completely in an additional 7 second stage. See Figure 10 showing the maneuver in two phases and the effect on the reduction of the flowrate through one of the valves. This solution was completely satisfactory: pressure did not go beyond the specified limits, the closure time was considered reasonable short, and the valve maneuver was technically possible using a two-speed motorized actuator.

Figure 10. Two-phase maneuver and effect on the flowrate through the valve

As said, the designed maneuver should be checked for other (less severe) scenarios. For example, during the night the consumption is much lower and, as a result, the transient more lenient. See the maximum and minimum enveloping piezometric grade lines for this situation in Figure 11.

Since presenting the platform architecture in full and/or a complete project is outside the scope of this paper, the paper has only briefly presented an overview of the most important elements such as the specification of agents and their behaviors, the establishment of the schedule, and the observation activities that may be developed through the platform. Such important aspects such as scenarios, groups of agents, accessible databases, and a more detailed description of the IDE have been left aside.

Figure 11. Max and min piezometric grade lines for closure in night conditions

Discussion on validation, scalability, computational complexity and applicability

The structure described in the paper is currently implemented in a software package (DiagastIng: http://fluing.upv.es/diagasting.php) with about fifty licenses (distributed mainly in Spain and South America). The package is used on a daily basis by practitioners to design water hammer protection strategies, and the results are reported to be excellent. As said, the network used in the previous paragraph corresponds to one of the various real-world case studies the authors have performed in recent years.

The calculation routines are based on a previous software package (DYAGATS (Izquierdo et al. 1996): http://fluing.upv.es/dyagats.php) developed by the authors – which has been in use since

1990 and has more than 300 current licenses. This package for the simulation of hydraulic transients in simple pipes has been applied extensively by many practitioners mainly in Spain and Latin America, and undergone numerous quality assurance tests. As a consequence, the authors claim that the calculations and results provided are reliable.

To build the platform four objectives were pursued. Firstly, the development of an infrastructure to handle complex hydraulic networks that understands hydraulic network topologies and can import (EPANET: Rossman 2002) and shapefile format files. Secondly, the endowment of such infrastructure with the know-how accumulated by the authors during more than two decades of experience with hydraulic transients. Thirdly, the incorporation of the main concepts behind multi-agent based applications to give the platform an MAS-orientated structure able to facilitate interesting advances and a distributed and parallelized standpoint. And fourthly, the implementation of the whole system using one of the most modern and efficient computer infrastructures, namely, .NET by Microsoft ©.

The implementation is robust (Huhn et al. 2003) in the sense that a reasonable variation in input does not take the algorithm out of control (note that unreasonable variations, mainly due to user errors, are virtually controlled). The various numerical algorithms used have been tested during years of use in the software package mentioned above. Traditional modeling of water hammer analysis has involved exhaustive prediction of all operating conditions, and this typically results in fragility for complex problems (Gribble 2001). The use of a MAB system avoids a detailed prediction of operating conditions.

An additional aspect relates to the consistency of agent interactions. Relations among agents have no problems of consistency within this system. Firstly, interaction between any two calculation nodes is clearly defined by (7) which is a set of linear equations with a unique solution. A similar argument may be used for interaction between pipe agents (through both ends) and consumption nodes, since this relationship is governed by equations (9) and also provides unique solution. Regarding all the other devices, consistency is well established since behaviors have been checked for years as very robust routines in DYAGATS. Finally, the facilitator relationship with any of the would-be interlocutors is perfectly stable, since it is consistently defined in terms of clear logical statements.

Regarding scalability, the authors claim that the underlying structure remains the same irrespective of any changes in system size. The only problem is posed by the computational capacity of the computer, or the system of computers where the package is run, since the more agents the greater the capacity needed. As a hydraulic network expands it is mainly the inner (calculation) pipe points that significantly increase in number. However, these agents are very simple. In fact, the behavior of any of these agents is described, as stated above, by a simple linear system of equations (7). As a result, the computational load of these agents does not increase with the size of the system. In addition, these agents are reactive per se. As a consequence, they only consume processor time when they compute a response to an incoming message. Proactive agents (mainly pumping stations) increase at a very small rate as the network expands. As a result, if scalability is measured in terms of the rate between performance and resources, the MAS presented in this paper exhibits a clear sub-linear scalability behavior – meaning that with the size of the network analyzed, the resources that are necessary do not impair system behavior.

Another important point is the following. Traditionally, given that waterhammer analysis needs high computational requirements, various problem simplifications are used. Such simplifications must be used with great care; otherwise they may produce results that are far from the real state of the system. If simplifications are reasonable, the results are perfectly acceptable from an engineering point of view. The use of a MAB system enables less simplification, thus reducing the risk of unsuitable simulations.

In any case, parallelizing the system will definitely help the scalability issue. If the number of concurrent threads can be distributed over different physical machines, it is possible to scale up the number of agents in a multi-agent system without risking a decreased performance by individual agents. In addition, fast and reliable message delivery is of the utmost importance to avoid potentially chaotic behavior. Distributing the load over multiple processors avoids an overload of message transmission.

Regarding computational complexity, the authors have to mention the following. Water hammer is a complex phenomenon described by a nonlinear set of PDEs. In addition, some of the elements constituting the boundary conditions exhibit great complexity, since they are described by sets of mixed algebraic-differential equations that need specific and refined numerical and computational methods; and, in addition, all the ingredients must be put to work in a synchronized manner to produce understandable results. The use of a MAB system enables coordinated interaction among the various elements constituting the system. As a result, a MAB approach is a very good option for reducing computational complexity problems.

Finally some facts related to applicability are provided. Focusing just on the field of expertise of the authors (namely, urban hydraulics), MAB methodologies, such as the one presented in this paper, have a wide range of applicability. Specifically, the authors have used MAB methods to build a tool to tackle the problem of the optimal design of water distribution networks using evolutionary algorithms that involve various self-adaptive types of agents that are able to fine-tune their parameters during evolution to improve performance (Montalvo et al. 2014). In addition, a MAB system has been developed to ‘sectorize’ a water supply network into DMAs, a problem of great interest in water supply management (Izquierdo et al. 2011, Herrera et al. 2011, 2012). In this case, agents are immersed in an energy-based negotiation process that enables an optimal division of the network into sectors, minimizing the number of cut-off valves necessary to isolate the various sectors. Various other MAB systems have also been devised by the authors, for example, one system analyzes images obtained by a GPR system when searching for hidden features of buried assets (pipes, valves, etc.) in a water supply system (Ayala-Cabrera et al. 2011, 2014).

CONCLUSIONS

Water supply is one of the more recognizable and important public services contributing to quality of life. Consequently, the security and integrity of this service must be guaranteed. One of the phenomena that puts in danger such a security is that of hydraulic transients. This is a very complex phenomenon (described by complicated models and solved by delicate numerical methods) that is difficult to visualize and interpret, and not easily predicted by simple judgments and decision-making. The computational implementation of the methodologies used to solve this problem is highly computer intensive. The calculation power and various capabilities of computers are sufficient to model such a phenomenon. Nevertheless, it is still necessary to consider certain aspects directly derived from the characteristics of the models that issue important warnings about such implementations.

This paper has presented a platform for the simulation of hydraulic transients in WDSs that uses a MAB system. 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 behavior and interacts with other elements. By means of the proposed multi-agent approach, combinations of simple elements in one location are implemented through new agents – called facilitators – that moderate the traffic of questions and answers among simple elements.

However, the biggest advantage in the approach that constitutes the base of the platform is that a system based on agents enables the parallelization of the calculation algorithm and this favors a better use of computer resources.

The presented platform is a tool of great interest for water supply managers, for whom transient analysis still represents a challenge in many cases. As an example, the inadequate protection of systems due to the excessively simplified analysis (Izquierdo et al. 2009, Jung et al. 2007) performed by most hydraulic transient simulation tools can be avoided by using the platform proposed in this paper. Moreover, overprotection (derived from adherence to excessively tight margins of safety, with the extra investment that this implies), for not having a sufficiently powerful tool can be avoided.

Finally, it is worth to mention that the whole application runs on a single computer. Ongoing research lines include parallelization of the algorithm. Although different groups of agents currently run on different threads, parallelization is a priority for future work since it will enhance some of the procedures in the package.

ACKNOWLEDGMENTS

The use of English was revised by John Rawlins.

REFERENCES

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

Ayala-Cabrera, D., Herrera, M., Izquierdo, J., and Pérez-García, R. (2011). "Location of buried plastic pipes using multi-agent support based on GPR images." *J. Appl. Geophys.*, 75(4), 679–686.

Ayala-Cabrera, D., Izquierdo, J., Montalvo, I., and Pérez-García, R. (2013). "Water supply system component evaluation from GPR radargrams using a multi-agent approach." *Math. Comput. Model.*, 57, 1927–1932.

Cao, K., Feng, X., and Ma, H. (2007). "Pinch multi-agent genetic algorithm for optimizing water-using networks." *Comput. Chem. Eng.*, 31(12), 1565-1575.

Chaudhry, M. H. (1986). *Applied Hydraulic Transients*, Van Nostram Reinhold, New York.

Cohn, R., Durfee, E., and and Singh, S. (2011)." Comparing action query strategies in semi-autonomous agents." *Proc., 10th Int. Conf. on Autonomous Agents and Multiagent Systems*, 3, 1287–1288.

Dibley, M. J., Li, H., Miles, J. C., and Rezgui, Y. (2011). "Towards intelligent agent based software for building related decision support." *Adv. Eng. Inform.,* 25(2), 311-329.

Evangelisti, G. (1969). "Waterhammer Analysis by the Methods of Characteristics." *L'Energia Elettrica*, nº 10, 11, 12.

Ghidaoui, M.S., Zhao, M., McInnis, D. A., and Axworthy, D. H. (2005). "A Review of Water Hammer Theory and Practice." *Appl. Mech. Rev*., 58, 49-76.

Gribble, S. D. (2001). "Robustness in Complex Systems." *Proc., Eighth Workshop Hot Topics in Operating Systems*, 21-26.

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

Herrera, M. (2011). *Improving water network management by efficient division into supply clusters*, PhD thesis, Universitat Politècnica de València, Valencia, Spain.

Herrera, M., Gutiérrez-Pérez, J. A., Izquierdo, J., and Pérez-García, R. (2012). "Combining multiple perspectives on clustering. Node-pipe case in hydraulic sectorization." *Int. J. Complex Syst. Sci.*, 2(1), 17-20.

Huang, W., Zhang, X., and Wang, J. (2011). "Multi-agent system computing and simulation of inter-basin water transfer." *Intel. Autom. Soft Comput.*, 17(7), 897-908.

Huhns, M. N., Holderfield, V. T., and Zavala Gutierrez, R. L. (2003). "Achieving Software Robustness via Large-Scale Multiagent Systems". *Software Engineering for Large-Scale Multi-Agent Systems*, García, A. et al. (eds.), 199-215, Springer Berlin Heidelberg.

Igual-Herrero, F. J. (2012). *An electronic institution based on Multi-Agent Systems for a water right transfer market in Spain*. Universitat Politècnica de València.

Izquierdo, J., and Iglesias, P. L. (2002). "Mathematical Modeling of Hydraulic Transients in Simple Systems." *Math. Comput. Model.*, 35, 801-812.

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

Izquierdo, J., Pérez, R., and Iglesias, P. L. (2004). "Mathematical models and methods in the water industry." *Math. Comput. Model.*, 39, 1353–1374.

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

Izquierdo J., Montalvo I., Pérez-García, R., and Izquierdo F. J. (2012). "Hydraulic transient simulation in networks using a multi-agent based approach." *Water Distribution System Analysis 2010 – WDSA2010*, Tucson, AZ, USA, Sept. 12-15, 2010.

Izquierdo, J., Herrera, M., Montalvo, I., and Pérez-García R. (2011). "Division of water supply systems into district metered areas using a multi-agent based approach." *Software and Data Technologies*, *Communications in Computer and Information Science*, Springer, Berlin, Heidelberg, Ch. 13, 167–180.

Izquierdo, J., Montalvo, I., Pérez-García, R., and Matías, A. (2012). "On the Complexities of the Design of Water Distribution Networks." *Math. Probl. Eng.*, 2012, 1-25.

Jung, B. S., Karney, B. W., Boulos, P. F., and Wood, D. J. (2007). "The need for comprehensive transient analysis of distribution systems." *J. AWWA,* 99(1), 112-123.

Kotina, R., Maturana, F. P., and Carnahan, D. (2006). "Multi-agent control system for a municipal water system." *Proc., 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, Madrid, Spain, 464-469.

Marchal, M., Flesh, G., and Suter, P. (1965). "The calculation of waterhammer problems by means of digital computer." *Proc., Int. Symp. on Water hammer in pumped storage projects,* ASME, 168-188.

Maturana, F.P., Kotina, R., Staron, R., Tichý, P., and Vrba, P. (2006). "Agent-based water\waste water control system architecture", *Proc., IADIS Int. Conf. Applied Computing*.

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

Montalvo, I. (2011). *Diseño óptimo de sistemas de distribución de agua mediante Agent Swarm Optimization*, Ph.D. thesis, Universitat Politècnica de València, Valencia, Spain.

Montalvo, I., Izquierdo, J., Pérez-García, R., and Herrera, M. (2014). "Water Distribution System Computer-aided Design by Agent Swarm Optimization." *Comput. Aided Civil Infrastr. Eng.*, 29(6), 433-448.

Nguyen, V. V., Hartmann, D., and Konig, M. (2012). "A distributed agent-based approach for simulation-based optimization." *Adv. Eng. Inform.*, 26(4), 814-832.

Nichita, C., and Oprea, M. (2007). "Water Pollution Diagnosis with a Multi-Agent Approach." *Proc., Artificial Intelligence and Soft Computing*.

Rossman, L. A. (2000). *EPANET 2 User's Manual*, Cincinnati (IN), USA, Environmental Protection Agency.

Ruiz, N., Giret, A., Botti, V., and Feria, V. (2014). "An intelligent simulation environment for manufacturing systems." *Comput. Ind. Eng.*, 76, 148-168.

Spellman, F. R., (2013). *Handbook of Water and Wastewater Treatment Plant Operations*, Third Edition, CRC Press.

Stone, P., and Veloso, M. (2000). "Multiagent Systems: A survey from a Machine Learning perspective." *Autonom. Robots*, 8(3), 345-383.

Thorley, A. R. D. (1991). *Fluid Transients in Pipeline Systems*, D. and L. George Ltd., England.

van Oel, P. R., Krol, M. S., Hoekstra, A. Y., and Taddei, R. R. (2010). "Feedback mechanisms between water availability and water use in a semi-arid river basin: A spatially explicit multi-agent simulation approach." *Envir. Model. Softw.*, 25(4), 433-443.

von Riemann, B. (1869). *Partielle Differentailgleichungen*, Ed. Braunwnschweig.

Weiss, G. (Ed.) (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*, MIT Press, 619.

Wooldridge, M. (2002). *An Introduction to Multiagent Systems*, John Wiley and Sons.

Wylie, E. B., and Streeter, V. L. (1993). *Fluid Transients in Systems*, Prentice Hall, Englewood Cliffs, N. J.

Yang, Y., Zhao, J., and Cai, X. (2010). "A Multi-Agent System Based Model for Water Allocation Management in the Yellow River Basin." *Proc., World Environmental and Water Resources Congress 2010*: 2656-2663.

Yazdani, A., and Jeffrey, P. (2011). "Complex network analysis of water distribution systems." *Chaos*, 21, 016111(2011).

Zhang, Y., Huang, G. Q., Sun S., and Yang, T. (2014). "Multi-agent based real-time production scheduling method for radio frequency identification enabled ubiquitous shopfloor environment." *Comput. Ind. Eng.*, 76, 89-97.

Zhao, T., Ding, R., and Man, Z. (2011). "Long-range monitoring system of irrigated area water-use based on Multi-Agent." *Proc., Int. Conf. on Mechatronic Sci., Electric Engineering and Computer (MEC)*, Jilin, China, 580 – 583.

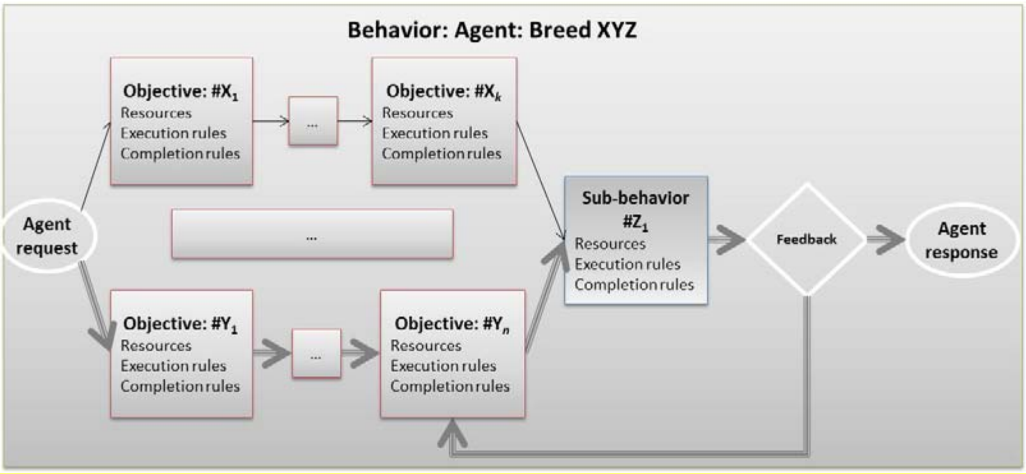


Figure 1. Example of agent behavior

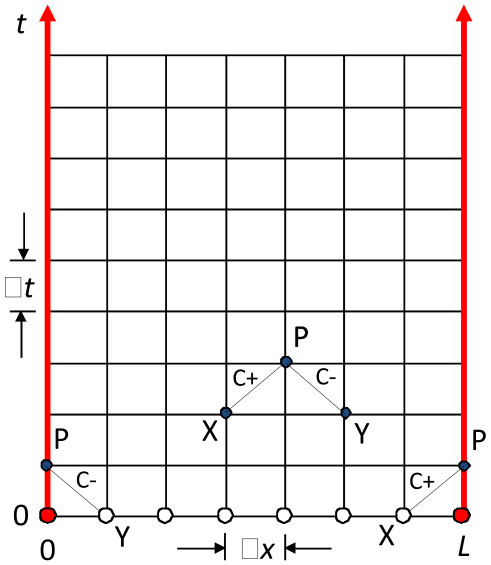


Figure 2. MOC discretization, characteristics, calculation points and boundary conditions

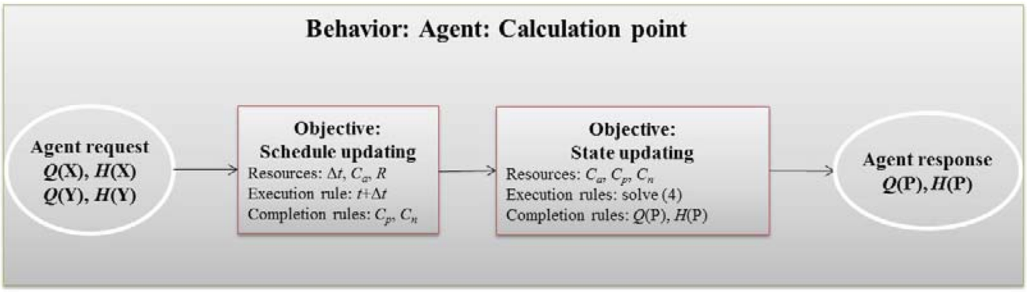


Figure 3. Behavior directed graph for a calculation point

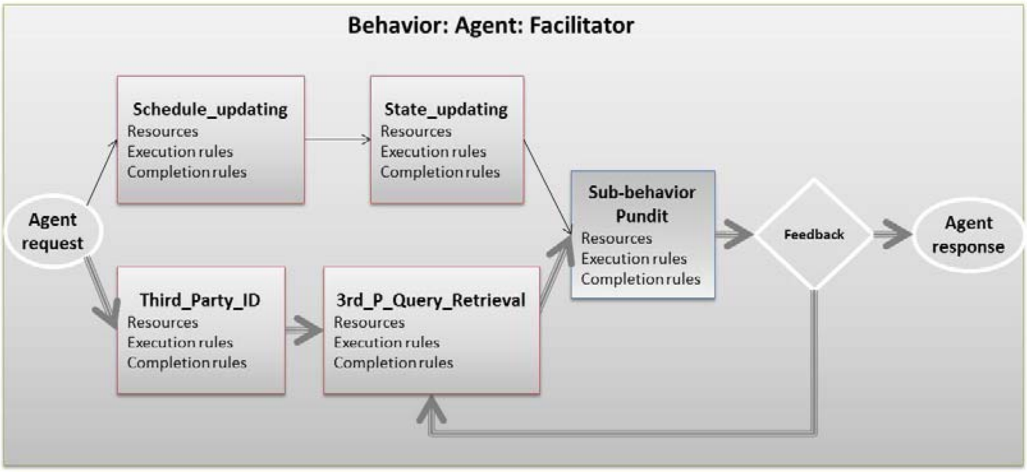


Figure 4. Facilitator behavior

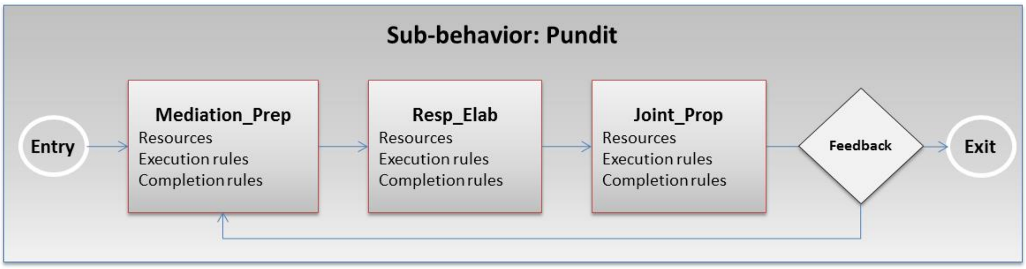
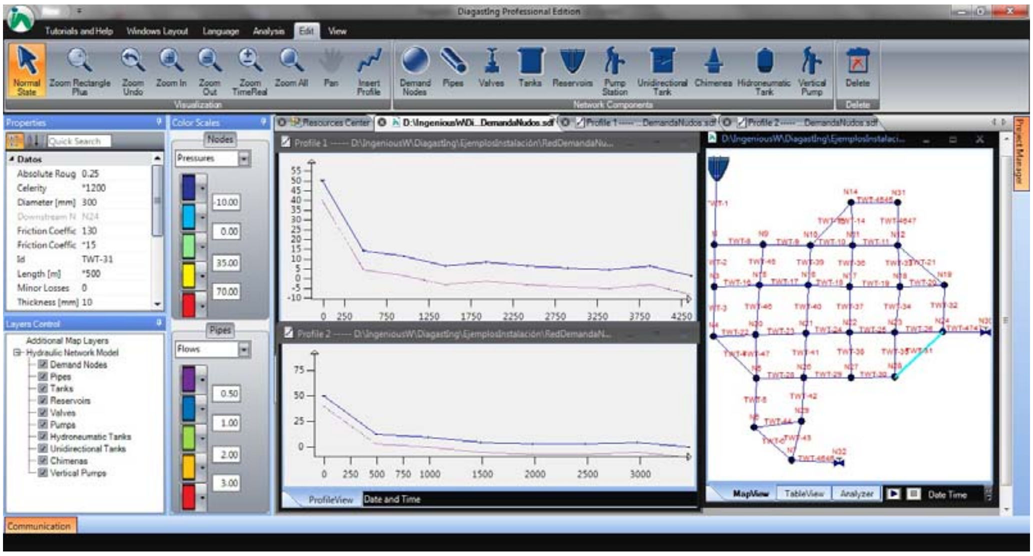


Figure 5. Sub-behavior of the facilitator



Datos

Absolute Roug	0.25
Celerity	*1200
Diameter [mm]	300
Downstream N	N24
Friction Coeffic	130
Friction Coeffic	*15
Id	TWT-31
Length [m]	*500
Minor Losses	0
Thickness [mm]	10

Nodes: Pressures — -10.00, 0.00, 35.00, 70.00

Pipes: Flows — 0.50, 1.00, 2.00, 3.00

Additional Map Layers
- Hydraulic Network Model
  - ☑ Demand Nodes
  - ☑ Pipes
  - ☑ Tanks
  - ☑ Reservoirs
  - ☑ Valves
  - ☑ Pumps
  - ☑ Hydroneumatic Tanks
  - ☑ Unidirectional Tanks
  - ☑ Chimenas
  - ☑ Vertical Pumps

Figure 6. Platform IDE

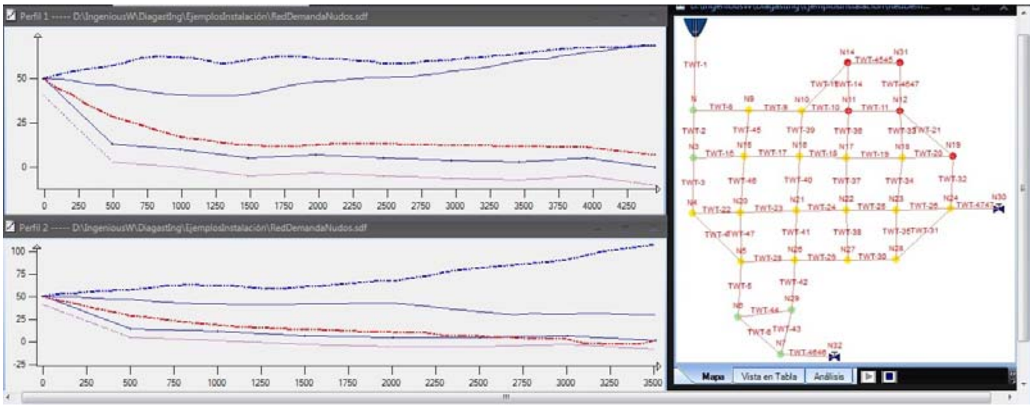


Figure 7. Snapshot of platform IDE showing grade lines on the profiles and color-coded points regarding the network view