

Article

The Organisational Structure of an Agent-Based Model for the Management of Wastewater Systems

Luis Oliva-Felipe ^{1,*} , Marta Verdaguer ² , Manel Poch ² , Javier Vázquez-Salceda ¹  and Ulises Cortés ¹ 

¹ Knowledge Engineering and Machine Learning Group (KEMlg@IDEAI), Universitat Politècnica de Catalunya—BarcelonaTECH (UPC), C/Jordi Girona, 1-3, E-08034 Barcelona, Spain; jvazquez@cs.upc.edu (J.V.-S.); ia@cs.upc.edu (U.C.)

² LEQUIA, Institute of the Environment, Universitat de Girona (UdG), C/Maria Aurèlia Capmany, 65, E-17003 Girona, Spain; marta.verdaguer@udg.edu (M.V.); manuel.poch@udg.edu (M.P.)

* Correspondence: loliva@cs.upc.edu

Abstract: Water managers have to deal with complex problems due to the intertwined characteristics of processes, in particular those that occur in wastewater systems. Existing modelling approaches are usually centred in the physical, chemical and biological aspects of the individual processes, excluding the social and organisational context that will generate the global behaviour. These also include the responsibilities and decision making of different actors in the system. This paper proposes an agent-based model with the novelty to integrate the social and organisational structure of the wastewater system from which emerges the global behaviour of the system. The modelling process allows considering the legal regulations and the technical limits that would drive the decision making. The instantiation of the model, implementing a small system, evidenced the usefulness of this approach to manage the complexity of wastewater systems and its possible contribution to prevent environmental problems.



Citation: Oliva-Felipe, L.; Verdaguer, M.; Poch, M.; Vázquez-Salceda, J.; Cortés, U. The Organisational Structure of an Agent-Based Model for the Management of Wastewater Systems. *Water* **2021**, *13*, 1258. <https://doi.org/10.3390/w13091258>

Academic Editor: Laura Bulgariu

Received: 5 April 2021

Accepted: 27 April 2021

Published: 30 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: wastewater systems; decision support systems; knowledge management; modelling

1. Introduction

The management of wastewater systems shows notorious importance in the water cycle due to its impact on the quality and the ecological state of rivers. The river often acts as a water supplier of the potabilisation plants, so poorly treated wastewater will require to a greater potabilisation effort. To manage the high complexity of wastewater systems, Rendón-Sallard et al. [1], Poch et al. [2], Cortés et al. [3], Comas et al. [4] and Benedetti et al. [5] proposed the use of decision support systems. In this context, the introduction of autonomous decision support systems, defined as agents (Wooldridge and Jennings [6]), has opened a new way in management systems (Cortés and Poch [7]). This approach allows modelling the interactions between components and stakeholders and modelling the different variable scales of the domain elements. Modelling and simulating such an integrated system is a robust means of transposing the knowledge of that system into predictions of it and will bridge the existing gap between theory and practice (see [1]).

An agent is an autonomous program capable of taking its own decisions based on its perceptions and its goals. A multi-agent system (MAS) allows representing real systems using a bottom-up approach: first, modelling the individuals and their social model; and then, letting the complexity of the system to emerge from the effects produced by their behaviours and interactions.

The literature shows different methodologies for developing a multi-agent system. Several widely referenced are examples include Gaia [8,9], MaSE [10,11], Tropos [12] and Prometheus [13]. The analysis of these methodologies shows that all (except for Prometheus) present difficulties for considering different abstraction levels [14]. However, as Zambonelli remarks in [15], generic MAS methodologies pose difficulties when developing real and

complex applications. Indeed, the performance characteristics of wastewater systems, with the changing characteristics of influent, effluent, and receiving waters, in addition to new wastewater inputs entering the system with wastewater being retained and treated by infrastructures from different stakeholders, can be partially modelled as the result of the interaction of different actors. However, modern wastewater systems result from the interaction of several public and private stakeholders with clear responsibilities and complex interdependencies. A more expressive framework is required to model not only the actors' capabilities, actions and effects in the system, but also some social factors such as their individual and collective responsibility, the structure of interactions that take place in agreed patterns, social phenomena such as authority, power and influence, and the existence of regional, national and international regulations.

In this work, the authors have selected the *ALIVE* framework [16]. *ALIVE* is both an agent-oriented software modelling framework and a methodology for the design of distributed computational systems. It allows designing and implementing systems, taking into account both (1) high-level organisational and coordination perspectives and (2) low-level, service perspectives [17]. In the present approach, services can be both digital Web-services operating in the Cloud (such as information services or other computational services) or connectors to physical systems (such as sensors and actuators or even interfaces with humans). *ALIVE* is especially well suited for highly regulated scenarios where changes can occur at either abstract or concrete levels, and where the list of services to be coordinated may change, with new services entering the system and existing services becoming unavailable at run-time [18]. *ALIVE* has been applied both for the dynamic orchestration of distributed, online services, and even to support the dynamic, flexible coordination of human teams in crisis management scenarios [18]. The model outlined in this paper is an extended and significantly expanded version of the preliminary work presented in [19].

2. Materials and Methods

The MAS model focuses on the individual and collective responsibilities of every stakeholder of the real system, and how they depend on others to achieve individual and collective goals, to ensure an adequate global performance. The key elements which describe the organisational structure of such a system are the roles. The roles group the activities necessary to achieve organisational objectives and clearly define who shall be responsible for those actions and the dependencies among them. Hence, dependencies among those activities are solved by means of coordinated action. An essential feature of role-based MAS models is that they decouple specific actors from the roles they enact [18]. The *ALIVE* framework describes an iterative process to use this approach to build the model, composed of the following five steps:

- Detect and name the stakeholders of the system;
- Formalise roles defining them, listing their goals, who depends on whom and who is responsible for what;
- Describe the interactions between the roles to handle their dependencies in terms of interaction scenes;
- The interactions defined in the previous step have to be linked into a meaningful interaction structure;
- Clearly state how agents are going to enact the different roles they have to play during the execution of the model.

The model was built using the OperettA editor (integrated in the *ALIVE* framework editor) and allows the construction of the Opera organisational model, which contains the elements described in this paper [20,21].

3. Modelling Process

3.1. Organisational Structure

The *ALIVE* methodology describes two main models: the *Organisational Model* and the *Social Model*. The *Organisational Model* includes:

- The *Social Structure*, which defines the roles and how they relate with each other.
- The *Interaction Structure*, which describes how the roles interact and how they coordinate in terms of scenes and landmarks to achieve.
- The *Communication Structure*, which lists the messages that agents should use—its kinds and purposes, and what words can be used as their content (i.e., pragmatics and semantics).
- The *Normative Structure*, which represents the norms that state what behaviours are acceptable, what is permitted and forbidden.

The *Social Model* explains what roles are played by each particular agent of the system, making explicit how roles are grouped and instantiated in the run-time system.

The following sections focus on presenting the *Organisational Model* (*Social Structure* and *Interaction Structure*), as well as the *Social Model*. It is important to note here that our model does not include a *Communication Structure*, since the purpose of our model was to implement simulations whose agents are not developed by different parties and therefore, communication among them does not require a complex model to describe an agreement, or explicit convention or shared knowledge about the terms to be used in messages. Moreover, the *Normative Structure* is not presented in this paper; although norms are an important component of institutions, either formal (e.g., UWS) or informal (e.g., neighbourhood groups), it depends on the specific legislation of its deployment. Nevertheless, the addition of a *Normative Structure* is planned as future work (see Section 5) to instantiate the model in a European Mediterranean basin.

This organisation should abstract the case study features, which are described in the next section.

3.1.1. Case Study: The Wastewater System

The wastewater system contains different elements which generate wastewater such as households (\mathcal{H}), industrial activities (\mathcal{I}), and meteorological events. Each component can (individually) generate wastewater with different dynamic and compositional characteristics. Wastewater characteristics include the volume (V) and the pollutant concentrations, with total suspended solids (TSS), biochemical oxygen demand (BOD), chemical oxygen demand (COD), total nitrogen (TN) and total phosphorus (TP) [22]. These pollutant concentrations are represented with the term \vec{C} , which denotes a vector of these pollutant concentrations where C_r , with $r = 1, \dots, 5$, denotes each specific pollutant concentration.

The system also has elements treating wastewater (Wastewater Treatment Plants $WWTP$), and receiving waters (e.g., a river, lake). Figure 1 shows the schema of flows for a hypothetical urban wastewater system.

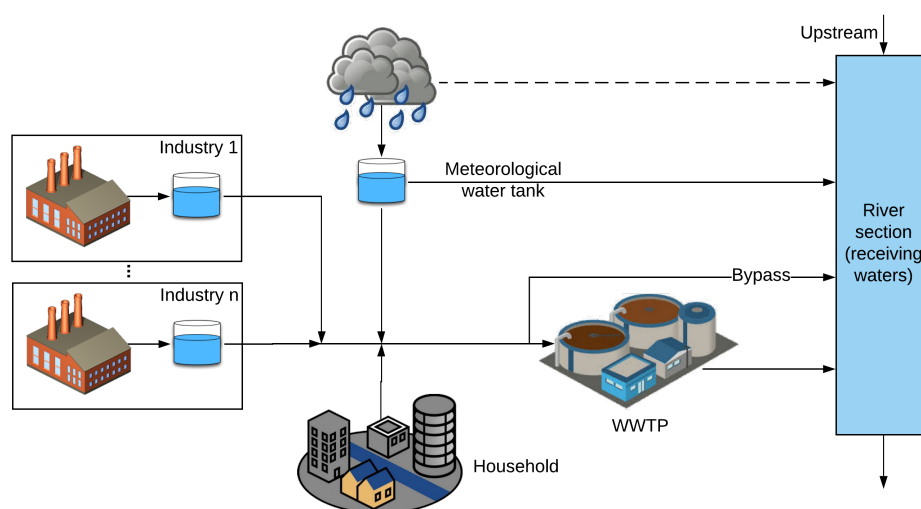


Figure 1. River basin case study.

3.1.2. Social Structure

The *Social Structure* abstracts the case study's roles, the relations among them, thus linking individual objectives and the society goals. A wastewater system's organisation's primary goal is to discharge an acceptable effluent (i.e., with low pollutant concentrations) for the receiving waters. Defining the roles allows encapsulating the stakeholders' responsibilities (see Figure 2, the roles identified, their objectives and sub-objectives).

Still, they also require specifying the dependencies between these roles to fulfil their goals. Some objectives may require another role for its fulfilment—the role interaction structure can model this kind of dependency on. It is possible to distinguish tree types of role dependencies:

- *Hierarchical*: one of the roles acts as an authority over others and therefore, whenever a superior requests something to its related inferior, the latter has to comply with that request. These are detailed as *H* in Figure 2;
- *Network*: when the roles depending on each other coordinate as peers to achieve a shared objective. This dependency is marked using an *N* in Figure 2;
- *Market*: when some roles act as producers and consumers, the former being providers of services or products at a given price; it is shown as *M* in Figure 2.

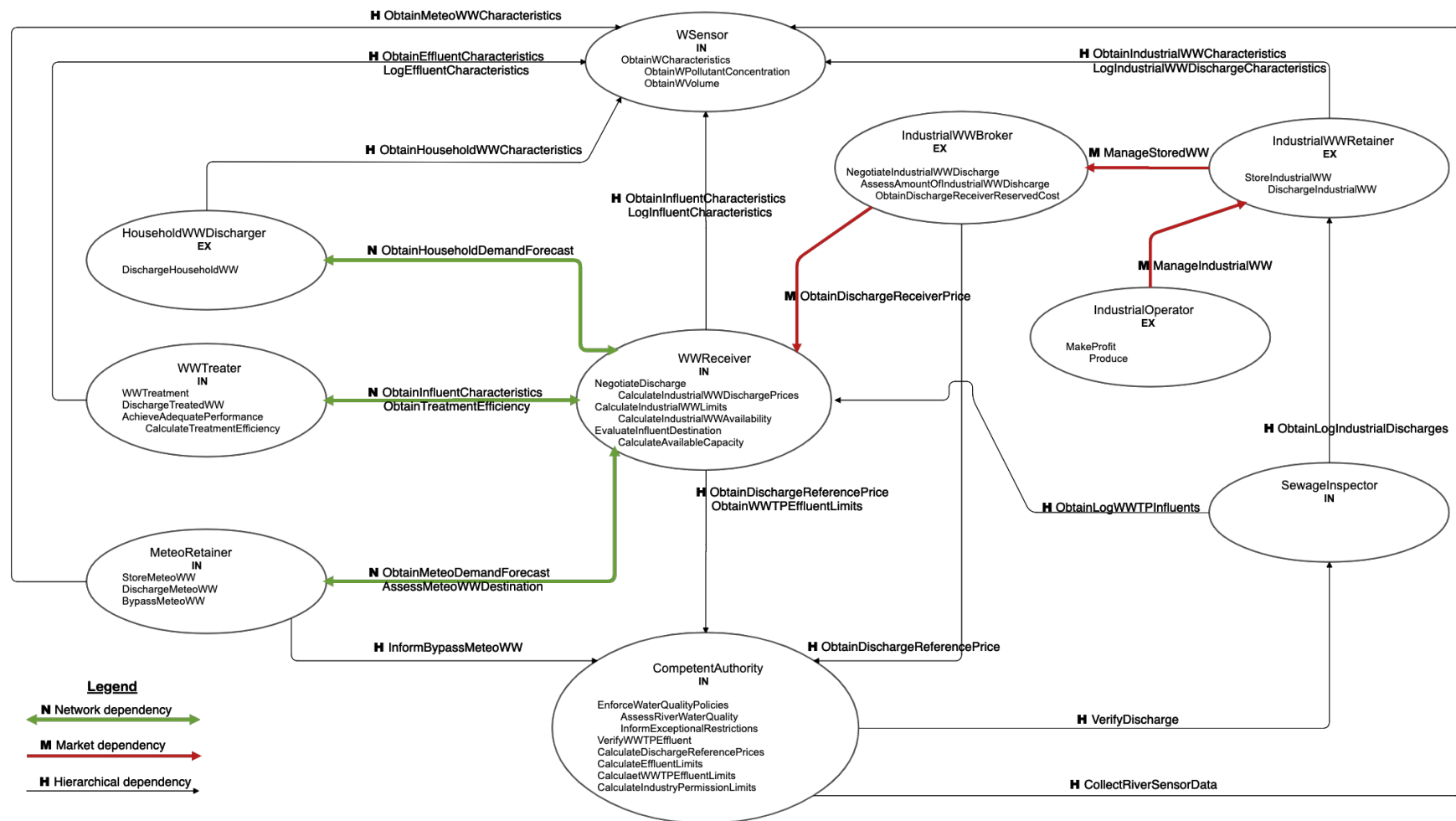


Figure 2. Diagram of the roles and role dependencies.

Roles can also be described as external or internal (detailed as *Ex* and *In*, respectively, in Figure 2) In the case where software components enact an internal role, this implies that it is possible for the organisation to manage its implementation (e.g., verify and validate it). In the model presented, the authorities responsible for wastewater management (or, indirectly, their computational systems) are internal roles. Contrary to internal roles, external roles are not controlled by the organisation despite being participants in it. If those external roles were implemented by software, it would imply that the organisation might not be able to check the code or validate its internal processes. In our scenario, external roles correspond to, e.g., the water generators.

In Appendix A, a complete list of the different roles is presented. An example of these roles was presented next (roles are presented in **bold** and goals in *italics*):

IndustrialOperator: role enacted by actors whose primary goal is to generate profit (*MakeProfit*) performing industrial processes, as a collateral effect of that industrial activity (*Produce*). This role, to manage the resulting wastewater depends on an **IndustrialWWRetainer** which can store wastewater for a subsequent discharge (*StoreIndustrialWW*), if required.

IndustrialWWRetainer: the main purpose of this role is to store industrial wastewater generated by an **IndustrialOperator** (*StoreIndustrialWW*) and manage its content properly (*ManageStoredWW*), by deciding to keep it stored or release it (*DischargeIndustrialWW*). Wastewater discharge depends on the **IndustrialWWBroker** role, to arrange such discharge and its cost, and evaluate whether it is possible to do a full or partial discharge, or just hold it for later release. If a discharge is performed, **IndustrialWWRetainer** logs it (*LogIndustrialWWDischargeCharacteristics*), so a register is built to be used by **SewageInspector** role. This role would then verify that everything is working as expected (*VerifyDischarge*). Logging discharges requires knowing its characteristics, thus creating dependency on the **WSensor** role.

IndustrialWWBroker: its responsibility is to negotiate how much industrial wastewater can be released and the cost of doing so. The negotiation takes place with a **WWReceiver** (*AssessAmountOfIndustrialWWDischarge*). This process depends on how much I_i is willing to pay (*ObtainDischargeReservedCost*) given the **WWReceiver**'s price.

IndustrialWWBroker: this role is responsible for negotiating industrial wastewater discharges with a **WWReceiver** to assess how much wastewater can be discharged (*AssessAmountOfIndustrialWWDischarge*). From the **IndustrialWWBroker** perspective, this assessment requires knowing its reserved cost (*ObtainDischargeReservedCost*) (i.e., how much I_i is willing to pay according to the discharge price given by the **WWReceiver**).

The **IndustrialWWBroker** will compute this reserved cost depending on the aforementioned discharge price requested to the **WWReceiver** (*ObtainDischargeReceiverPrice*):

- The reference prices are given by the **CompetentAuthority** with regards to volume and concentration of discharges (*ObtainDischargeReferencePrice*);
- The characteristics of the wastewater (i.e., volume and concentrations of pollutants) to be discharged, which are provided by the **IndustrialWWRetainer** as part of the wastewater discharge negotiation process.

Once negotiation ends, either one of the following may happen:

- An agreement is not achieved, and wastewater is kept stored until prices go down (if storage capacity allows doing so);
- An agreement is achieved, and some (or all) wastewater is discharged; the actual discharge may comply or not to what was previously agreed.

Figure 2 shows the dependency relation between the **IndustrialWWBroker** and the **WWReceiver** is a market dependency (*M*). Usually, a **WWReceiver** will also work with a **WWTreater** role to provide treatment capacity for discharged wastewater and therefore, when received wastewater is discharged later on, it will comply with the legal constraints that affect those discharges and guarantee a suitable water quality when it reaches receiving

waters. The **IndustrialWWBroker** is consuming the service provided by actors enacting the **WWReceiver** role.

WWReceiver: this role is responsible for arranging the arrival of industrial wastewater discharges (*NegotiateDischarge*). This negotiation depends on having the discharge prices for \mathcal{I} (*CalculateIndustrialWWDischargePrices*) and knowing how much treatment capacity is available (*CalculateIndustrialWWAvailability*). To determine wastewater discharge prices, some information is needed:

- Reference prices from the **CompetentAuthority**;
- The treatment efficiency of $wwtp_j$ (*ObtainTreatmentEfficiency*) and;
- The characteristics of the wastewater whose discharge is being negotiated by an **IndustrialWWRetainer**.

In addition, treatment capacity availability requires knowing:

- How much household wastewater is being received (*ObtainHouseholdDemandForecast*);
- Whether there is any meteorological phenomena expected (*ObtainMeteoDemandForecast*);
- Whether the **CompetentAuthority** has imposed any limitations on the $wwtp_j$ effluent (*ObtainWWTPEffluentLimits*);
- Treatment efficiency in the **WWTreater** (i.e., WWTP) (*ObtainTreatmentEfficiency*).

Once a discharge is arranged, it will eventually arrive. At the moment of the reception, **WWReceiver** decides its destination (*EvaluateInfluentDestination*). The wastewater can be accepted for treatment or, depending on current circumstances (e.g., unexpected rainfall), may be bypassed to the receiving waters. This decision is affected by the influent characteristics (*ObtainInfluentCharacteristics*), current $wwtp_j$ efficiency (*ObtainTreatmentEfficiency*) and how much space is available (*CalculateAvailableCapacity*). Similarly to **WWRetainer** or **WWTreater**, it logs received wastewater characteristics (*LogInfluentCharacteristics*). This information is required by the **SewageInspector** (to fulfil its *VerifyDischarge* goal) and it is also used to compute treatment efficiency.

WWTreater: a role responsible for decreasing the concentration of pollutants in wastewater (*WWTreatment*) before its release (usually into receiving waters such as a river). Similarly as in **IndustrialWWRetainer**, the effluent is analysed before its release (*ObtainEffluentCharacteristics*) and used for two reasons: to log it for any possible audit from the **CompetentAuthority** (*VerifyWWTPEffluent*), and to compute the treatment efficiency (*CalculateTreatmentEfficiency*) which, likewise, requires knowing the characteristics of the influent (*ObtainInfluentCharacteristics*). Knowing its efficiency facilitates the **WWTreater**'s task of pursuing its goal of being efficient (*AchieveAdequatePerformance*). Usually, this role is also needed to assist **WWReceiver** during the negotiation process with an **IndustrialWWBroker**.

3.1.3. Interaction Structure

The *Interaction Structure* allows the description of abstract patterns of interaction which are the way that the roles coordinate their behaviour, managing their dependencies while they pursue their individual and collective objectives. The interaction structure (see Figure 3) defines interaction patterns composed of a set of scenes and transitions among them [18]. Each scene represents a meaningful subset of the interaction between the system stakeholders, that is, an interaction pattern between two or more roles to coordinate their behaviours to achieve one or several (social) goals. On every scene, one or more role dependencies (identified in the previous phase) are managed. Table 1 summarises the role dependencies shown at every particular scene. Transitions mark the valid interaction paths for actors. Depending on the role(s) they enact, they may move from one scene to another. They may enter the scene at any time or only after meeting some conditions (e.g., there are enough actors from all roles to run the scene).



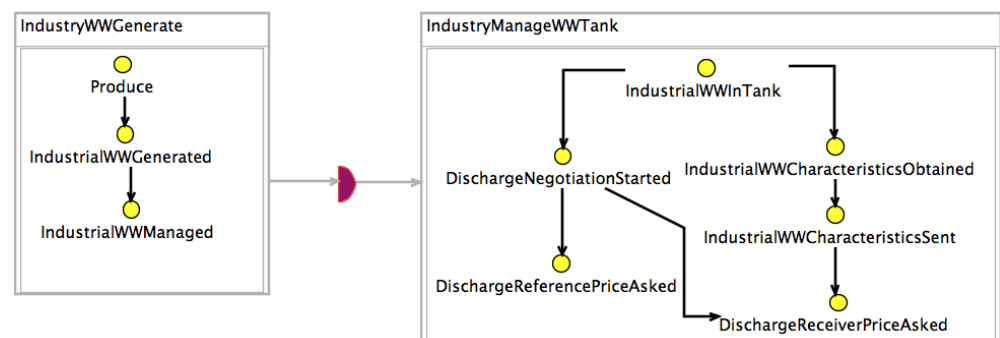
Figure 3. General scene pattern. It defines scenes that describe how agents coordinate and how transitions are performed from one scene to another.

Table 1. Scenes and role dependencies.

Dependency	Scenes
CollectRiverSensorData	EnforceWaterQualityPolicies
LogIndustrial WWDDischargeCharacteristics	ComputeWWTPEffluentLimits
LogInfluentCharacteristics	IndustryWWDDischarge
ObtainEffluentCharacteristics	WWTPReceiveInfluent
ObtainHouseholdWWCharacteristics	WWTPTreatInfluent
ManageIndustrialWW	MeteoWWGenerate
ManageStoredWW	IndustryWWGenerate
	IndustryManageWWTank
ObtainDischargeReferencePrice	IndustryAssessWWDDischarge
ObtainDischargeReceiverPrice	IndustryAssessWWDDischarge
ObtainWWTPEffluentLimits	IndustryAssessWWDDischarge
VerifyDischarge	WWTPAssessDischargePrice
ObtainTreatmentEfficiency	VerifyDischarge
ObtainInfluentCharacteristics	WWTPAssessDischargePrice
LogEffluentCharacteristics	WWTPTreatInfluent
ObtainEffluentCharacteristics	WWTPBypassInfluent
ObtainIndustrialWWCharacteristics	WWTPTreatInfluent
AssessMeteoWWDestination	WWTPAssessDischargePrice
	MeteoWWGenerateScene
ObtainIndustrialWWCharacteristics	IndustryManageWWTank
ObtainMeteoDemandForecast	WWTPAssessDischargePrice
ObtainHouseholdDemandForecast	WWTPAssessDischargePrice
ObtainLogWWTPInfluents	VerifyDischarge
ObtainLogIndustrialDischarges	VerifyDischarge
InformBypassMeteoWW	MeteoWWBypass

The structure's entry point is represented by a circle (*init* label), while the triangles represent exit points (*end* label). Rectangles represent scenes, and they are connected by lines (scene transition arcs) that allow the system to navigate from scene to scene. Scenes are connected through red semi-circles or green triangles to represent that the scene flow will move on once all scenes have been completed (for semi-circles), or at least one of them has been completed (for triangles). The landmark patterns describe the protocol used by actors to achieve the scene result [18].

Figure 4 shows the process of industrial wastewater discharge. The figure depicts the internal landmark patterns for the *IndustryWWGenerate* and *IndustryManageWWTank* scenes; as a result of the industry's production process (*Produce*), wastewater is generated (*IndustrialWWGenerated*) and stored in retention tanks. As retention tanks have limited capacity, they have to be managed (*IndustrialWWManaged*) to determine the best moment for releasing their contents into the WWTPs. To determine the cost of treating generated wastewater, whenever wastewater is available on the tank (*IndustrialWWInTank*), it is analysed (*IndustrialWWCharacteristicsObtained*). *I* also starts a negotiation with WWTP to determine the discharge price for wastewater reception (*DischargeNegotiationStarted*). The *I* also needs the reference price from the **CompetentAuthority** (*DischargeReferencePriceAsked*), to be used as a criterion. The negotiation requires the provision of wastewater characteristics to WWTP (*IndustrialWWCharacteristicsSent*), so the discharge receiver price can be effectively asked (*DischargeReceiverPriceAsked*).



<i>Description</i>	Scenes for industrial wastewater generation. Industrial wastewater is produced and temporarily stored while the industry agent negotiates with WWTP as to when to perform a discharge, and its price.
<i>Players</i>	IndustryWWGenerate: IndustrialOperator , IndustrialWWRetainer IndustryManageWWTank: IndustrialWWBroker , CompetentAuthority , IndustrialWWRetainer , WWSensor , WWReceiver
<i>Results</i>	IndustryWWManaged results in profit being generated for Industry and wastewater stored. IndustryManageWWTank results in discharge prices obtained (for both, the reference price from CompetentAuthority and the current price from a WWReceiver).

Figure 4. Composite of scenes for industrial wastewater generation and (partial) management.

See the landmark patterns for other processes in Appendix B.

3.1.4. Social Model

The *Social Model* describes which organisational roles play each agent. These roles drive agent's behaviour and support agents to coordinate by complying with the scenes defined in the *Interaction Structure*. An agent can play any number of roles; it depends on the goals of these roles. In addition, more than one agent can play the same role. For the

purpose of the case study, the UWS's objectives are the main criteria to assign roles. Figure 5 depicts the social model for the case study. The square-shaped elements are the agent types: *Household*, *WWTP*, *Industrial*, *River Council* and *Meteorological*. The round-shaped elements are the roles. The directed links represent which agents enact each role.

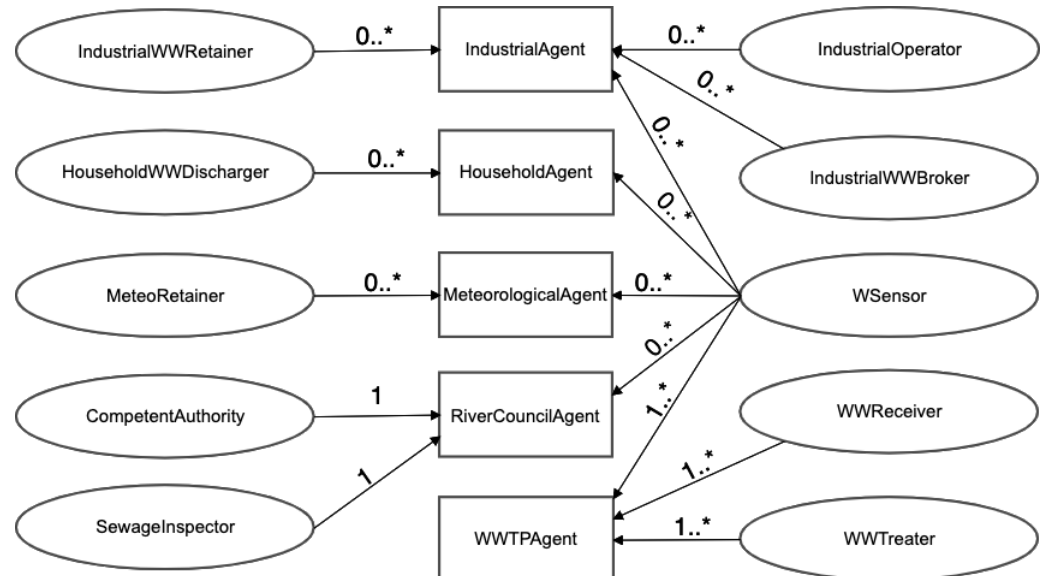


Figure 5. The social model defines the roles and the different types of agents that can play them. The * means that the amount of possible agents enacting this role can be many.

3.2. Behaviour and Decision-Making

In previous model (*Social Model*), the *RiverCouncil* agent is in charge of affairs concerning the collection, treatment and disposal of wastewater. The *WWTP* agent represents a given $wwtp_j \subset WWTP$, who has to reduce pollutant concentration in wastewater before releasing it. The release of wastewater needs an agreement between an industry and the *WWTP* that will receive it for treatment. For such an agreement to take place, $wwtp_j$ needs to know if it can properly handle the wastewater. The agreement process works as follows: $wwtp_j$ checks whether it can accept a discharge from I_i (a mass represented as (V_i, \vec{C}_i)). This means warranting there is sufficient volume capacity available (volume availability) and that the *WWTP* can treat the wastewater pollutants (pollutant concentration admissibility). The volume available is determined (see Equation (1)) by the design volume of the plant ($V_{capacity}$) minus the amount of household wastewater (V_d) and meteorological forecasts (i.e., rain) (V_m), which the *WWTP* is obliged to accept. It also has to take into account any agreements already made to other industries ($V_{scheduled}$).

$$V_{available} = V_{capacity} - V_d - V_m - V_{scheduled} \quad (1)$$

Then, if $V_i \leq V_{available}$, it means there is sufficient capacity in the *WWTP* to accept the industrial wastewater.

However, to achieve an agreement, it is also necessary that the plant can admit its pollutant concentration. This condition depends on how much pollutant concentration the plant can effectively cope with, which is determined by the plant's design parameter ($\vec{C}_{admissible}$). Although it is not something desired, sometimes admissibility may be overlooked; which implies that process will incur an additional cost, depending on the specific pollutant (C_r) and the amount exceeded. This will affect the price given by $wwtp_j$ accordingly (see Equation (2)):

$$P(WW_i, wwtp_j) = P((V_i, C_i), wwtp_j) = VC(V_i, wwtp_j) + PC(WW_i) \quad (2)$$

There are two elements in Equation (2): VC (volumetric cost) and PC (pollutant cost). On the one hand, the VC component represents the cost of allowing the entrance of a given wastewater volume (V_i) released by I_i given the status of $wwtp_j$. On the other hand, the PC component represents the cost of treating a given wastewater mass (WW_i), with certain pollutant concentrations, released by I_i . These costs could have additional taxes, which are imposed by the responsible authority.

Volumetric cost is defined as

$$VC(V_i, wwtp_j) = UC(V_i) \cdot \left(1 + \frac{V_{ocWWTP}}{V_{totWWTP}}\right)^x \quad (3)$$

The first part of Equation (3) is developed as

$$UC(V_i) = V_i \cdot [GT + ST(i)]; \quad (4)$$

GT corresponds to the general tax paid by industries while $ST(i)$ is the specific tax that depends on the type of industry where I_i belongs. This formula expresses the industry's taxes paid to discharge a specific wastewater volume (V_i). These taxes are defined by the competent authority. The second part of (3) is an exponential function that measures how much volume is available in the $wwtp_j$ from its total capacity. The potential factor, x , expresses the fact that the price increases as the $wwtp_j$ becomes fuller.

The following formula, based on [22], defines the pollutant cost:

$$PC(WW_i) = PC((V_i, C_i)) = V_i \cdot \left(\sum_{r=1}^5 C_i^r \cdot q^r \cdot g_i^r\right) \quad (5)$$

where:

- q^r is the reference price for discharging a mass unit of pollutant r and the competent authority determines it;
- g_i^r is a peak coefficient that expresses the deviation between the pollutant r concentration in the wastewater mass sent by the industry I_i and the pollutant concentration limits agreed with the competent authority.

The coefficient g_i^r allows industries to deal with unforeseen situations that force them to discharge more pollutant concentrations than they are entitled to, by paying a higher price for doing so. It is calculated as

$$g_i^r = \frac{C_i^r}{X_i^r}$$

where X_i^r corresponds to the maximum concentration of pollutant r in a discharge, agreed between industry I_i and the competent authority. In this way, Equation (2) considers the permits given by the responsible authority to industries in terms of allowed pollutant concentration and taxes for discharging wastewater, and the current status of $wwtp_j$ (in terms of volume occupancy). This will also enable industries to overpass pollutant concentration to deal with unforeseen situations as long as they can pay for the extra cost it implies.

Finally, $wwtp_j$ communicates the price and available volume to the industry. Then, I_i decides to either keep storing the wastewater in its internal storage tanks or to proceed with a full or partial discharge.

3.3. Negotiation

The negotiation protocol allows for I_i to negotiate with $wwtp_j$, in a *peer-to-peer* fashion, in order to reach an agreement for treating industrial wastewater in a decentralised way. This protocol is initiated by I_i , and will occur typically when their wastewater tanks are full, looking for the lower wastewater treatment price and therefore the maximum benefit, which is the ideal situation from I_i 's perspective. In addition, the protocol allows for $wwtp_j$

to negotiate with I_i providing offers for wastewater treatment. This scenario will typically occur when a plant's demand is low, with respect to the plant's treatment capacity, and will allow to any $wwtp_j$ to balance its treatment capabilities, offering to treat wastewater at a lower cost when their demand is lower in valley hours.

4. Results and Discussion

To check the model feasibility, the conceptualisation has been used to build an agent-based simulation. The simulation has been implemented using Repast Symphony [23]. The execution followed the indications depicted in the model:

- Linking goal-driven behaviours;
- Allowing them to choose which scene they will navigate to (i.e., discharge wastewater to a *WWTP* or store it in a retention tank), and how (which *WWTP* or tank to use in either case);
- Selecting among the set of actions available to transition from landmark to a landmark (within scenes) and from one scene to the next one.

In this way, the simulation can reproduce the behaviour of the most important stakeholders (e.g., industries, *WWTPs* or households) as well as the impact of their decisions and actions in the receiving waters, evidencing feasible environmental problems.

A complete execution trace of the simulation is presented in Appendix C. It represents a simple hypothetical scenario: no rainfall at all, a *WWTP* (with a storage capacity of 60.0 volume units) and two industries (Industry 1 and Industry 2, with a storage capacity of 40.0 and 20.0 volume units, respectively). This trace was focused on checking that interactions and processes modelled are performed correctly. This means having a deterministic scenario that verifies the correctness of expected interactions and protocols. In addition, a set of unitary tests have been implemented with the collaboration of domain experts. These tests include each atomic action (e.g., produce, store, discharge, water treatment) and the scenes that involve a *WWTP* and industry. Finally, an integration test that composes the flow of scenes that simulate the interaction between these two industries and a *WWTP* has also been coded. This test has been the basis to produce the trace. Figure 6 summarises the trace visually. The first ticks are for warming up the system, and after eighteen ticks, a pattern emerges. We set up the treatment process to last four ticks, and each industry produces ten-volume units of wastewater. This means that Industry 2 cannot store all wastewater, and it is spilling it (orange line, with up-pointing triangle symbol). As a result of this, the *WWTP* cannot hold off all the incoming water, and it has to bypass it to the river (the dark red line, with down-pointing triangle symbol, and the peaks in the dark blue line, with square symbol, which represent the volume being held in the basin). In contrast to this, Industry 1 is generally performing since it has enough storage capacity so the *WWTP* can treat water and allocate new space.

In addition to this simulation execution, the model has also been used to feed a norm-monitor system, NoMoDEI [24] Figure 7a is a snapshot of the norm-monitor screen showing a bird's eye view of the scene diagram. This includes information of what scenes have been selected by agents, that is, after agents performed their intended actions, which states of the world were achieved. Moreover, as seen in Figure 7b, it is possible to retrieve detailed information for each scene, such as the number of agents currently participating in that scene, how many agents have been engaged in it (history of past scene engagements), how many active norms have been instantiated, and how many of these norms are currently violated. Nodes are coloured as follows:

- Orange nodes stand for blocked scenes (i.e., scenes no agent has participated in, neither in the past nor at the present moment).
- Blue nodes stand for paused scenes, nodes that have been participated in by some agent in the past but not currently.
- Green nodes stand for active scenes, with actual participating agents.

Edges between nodes denote transitions between scenes. An edge contains a coloured-square at its destination side. Red squares represent a blocked scene transition (i.e., never been used by any agent), and blue ones represent active scene transitions (i.e., the agent has used them).

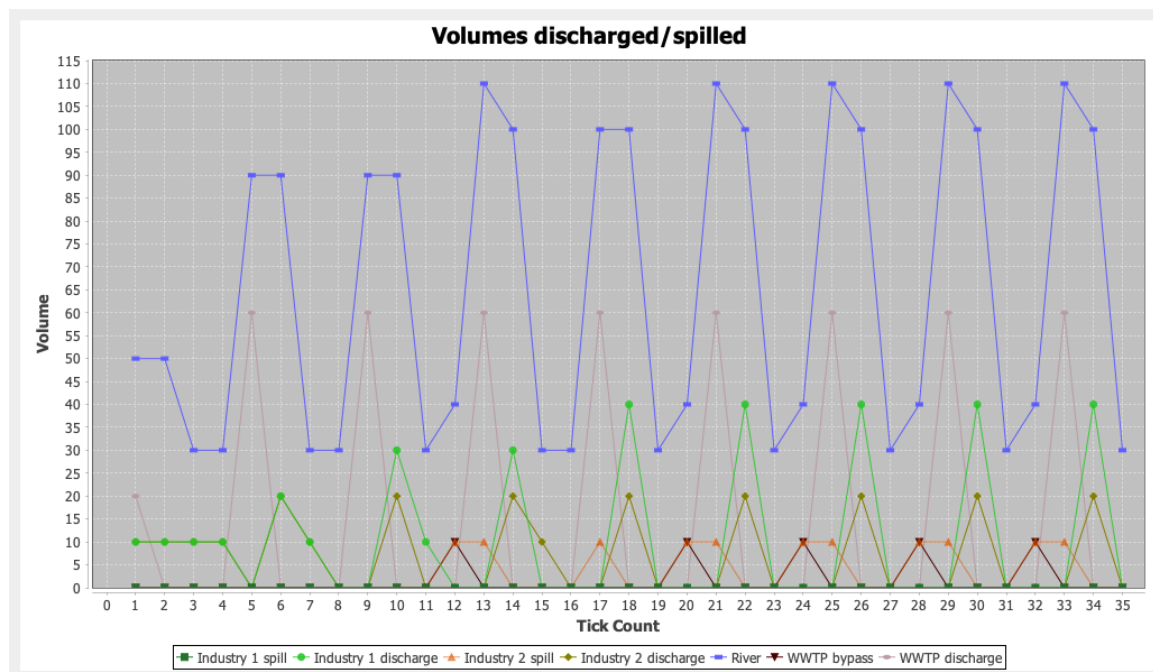


Figure 6. The figure shows an execution with one WWTP with an incoming water tank of 60 volume units and two industries (storage tanks with 40 and 20 volume units of capacity each). No rainfall.

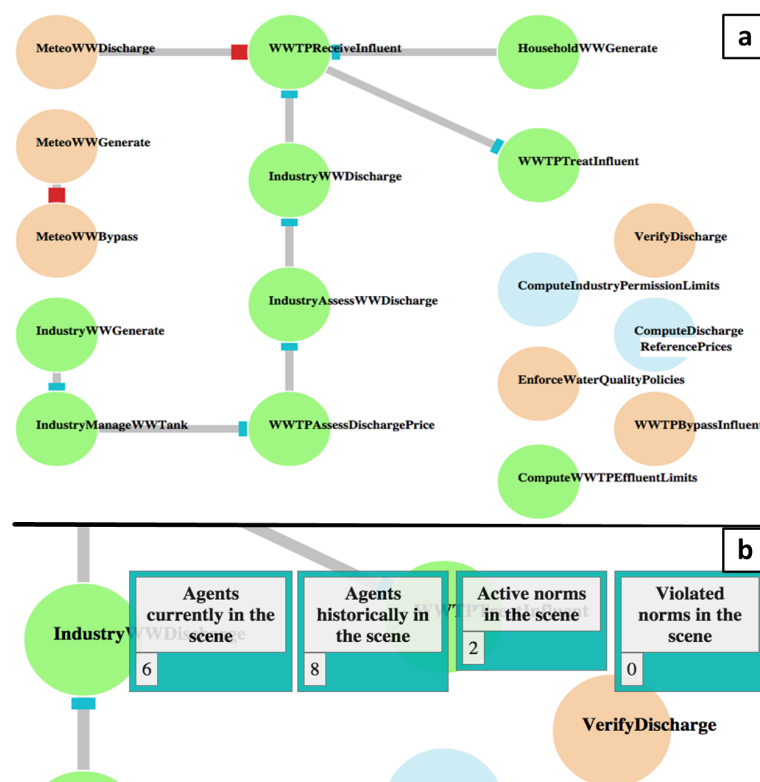


Figure 7. (a) Agents are traversing the different scenes during the simulation as they perform actions; (b) detailed information about the number of participating agents and which have played in a particular scene, i.e., *IndustryWWDischarge*.

This visualisation tool allows to contrast the social model with the dynamic simulation data, thus monitoring in real-time which scenes are taking place, which scenes ones are the most performed, or if some are never visited. This capability allows detecting potential design flaws in the social model or in the design of the agent's behaviour (i.e., via scenes that are never visited by agents). It is also possible to monitor norm activation and violation with regards to the scenes, that is, whether a scene is producing norm violations [24].

5. Conclusions

In this paper, we present a novel modelling approach for wastewater management systems that is able to model the expected behaviours of the overall system via the definition of the social and organisational responsibilities from different stakeholders and the patterns of interactions to be followed by the different stakeholders to deal with the water disposal and treatment in a distributed, effective way, and contribute to watershed's protection. Our model is based in *ALIVE*, a flexible methodology we used to model the complex interactions among different actors via the description of the roles they play, the goals they are expected to achieve and the dependencies on other stakeholders.

The *Social* and *Interaction Structures* presented in this paper (see Sections 3.1.2 and 3.1.3) allowed us to model a significant group of heterogeneous agents (representing different stakeholders in the water management system) with diverse individual responsibilities, interests, and behaviours and an emerging global behaviour. For each scenario, non-technical experts can assess the state of the abstract society that the system represents by checking the set of landmarks traversed by the different actors (see Figure 4 as an example; also see [24] and Figure 7 as an example on how the model presented is used to create a visual representation that eases logging the agents' behaviour in terms of the scenes). Moreover, using the proposed methodology, the user can easily modify the scenario to introduce new actors, augmenting its complexity. Thus, the methodology demonstrates its applicability at a regional scale and targeted at evaluating long-term scenarios. By using this methodology, both domain experts and policy makers can model the system as a whole, indicating the overall dependencies and the societal objectives of the stakeholders independently of their individual behaviours, thus allowing great flexibility to run different scenarios or integrate third-party implementations as long as they comply with the organisational model.

Furthermore, the MAS paradigm has demonstrated its usefulness in managing the complexity of wastewater systems and its feasible contribution to preventing environmental problems—in addition to being an approach applicable to other aspects of water management.

The *ALIVE* framework allows to enrich the *Organisational Model* with the definition of its *Normative Structure*, defining permissions, obligations and prohibitions, applied to role definitions or scene descriptions. The *Normative Structure* significantly depends on the local legislation of the place where the model was to be implemented. In this paper, we did not include a model of the *Normative Structure* for this scenario (but the interested reader can find examples of norms applied to an earlier version of the model presented in this paper in [19,24], showing how domain experts can easily add normative constraints relevant to their needs). We plan to perform a full normative analysis on a specific regulatory scenario, such as the Mediterranean basin in a European country. As a next step, we plan to do a normative analysis involving different legislative levels, including: European (Water Framework Directive), national (e.g., Spanish) and local (e.g., Catalonia). This analysis will allow to further refine the model by incorporating the body of regulations that would affect the roles, scenes and interactions in actual deployment.

Author Contributions: Conceptualization, U.C., M.P.; methodology, M.V., L.O.-F.; software, L.O.-F.; validation, L.O.-F., M.V.; visualization, L.O.-F.; writing—original draft preparation, U.C., M.P., L.O.-F., M.V., J.V.-S.; writing—review and editing, M.V., L.O.-F., U.C., J.V.-S.; supervision, U.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by Technical University of Catalonia (UPC).

Acknowledgments: LEQUIA has been recognised as consolidated research group by the Catalan Government with code 2017-SGR-1552. KEMLg@IDEAI is part of a research group, recognised as consolidated research group by the Catalan Government with code 2017-SGR-574.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Description of the Roles (Those of the System Not Specified in the Paper)

- **WSensor:** This role is focused on analysing the characteristics of a water body (e.g., a wastewater mass or a water mass in a river) (*ObtainWCharacteristics*). This analysis includes:
 1. Determining concentration of pollutants $\langle TSS, BOD, COD, TN, TP \rangle$ (*ObtainPollutantConcentrations*);
 2. Measuring the volume of water (*ObtainVolume*);
- **MeteoRetainer:** role responsible for storing water coming from meteorological conditions (e.g., rain) (*StoreMeteoWW*). This stored water is analysed (*ObtainMeteoWWCharacteristics*) and its destination is assessed (*AssessMeteoWWDestination*) (e.g., the storage tank is overflowed and it cannot retain more water). Depending on the assessment result, water is either:
 - Discharged to a **WWReceiver** (*DischargeMeteoWW*);
 - Bypassed to the river (*BypassMeteoWW*), which also requires informing the **CompetentAuthority** of such event (*InformBypassMeteoWW*);
 - Kept stored, if possible.

The relation between the **MeteoRetainer** and **WWReceiver** is a network dependency. Both roles act as equals since none of them have an authoritative power over the other nor they are consuming any service provided as in a market. This relation is motivated by the willingness to coordinate their actions and achieve the system's goal of providing a certain level of water quality which emanates from the **CompetentAuthority** goals.

- **HouseholdWWDischarger:** This role generates wastewater produced by residential areas and discharges it to the sewage system (*DischargeHouseholdWW*). It provides the wastewater characteristics that it generates to roles that may need it (*ObtainHouseholdWWCharacteristics*) such as **WWReceiver** as part of its goal *ObtainHouseholdDemandForecast*.
- **WWReceiver:** this role takes care of negotiating the reception of wastewater masses (*NegotiateDischarge*). This includes providing discharge prices for \mathcal{I} (*CalculateIndustrialWWDDischargePrices*) and the treatment capacity available for industrial wastewater (*CalculateIndustrialWWAvailability*). To calculate discharge prices, actors enacting the **WWReceiver** role use the discharge reference prices provided by the **CompetentAuthority**, the current treatment efficiency in $wwtp_j$ (*ObtainTreatmentEfficiency*) as well as the characteristics of the industrial wastewater that a given I_i wants to discharge, which are provided by the **IndustrialWWRetainer** as part of the wastewater discharge negotiation process. Concerning treatment capacity availability, it depends on:
 - The wastewater being received from households (*ObtainHouseholdDemandForecast*);
 - Meteorological retainers status (*ObtainMeteoDemandForecast*);
 - Current efficiency in the **WWTreater** (*ObtainTreatmentEfficiency*);
 - Current $wwtp_j$ effluent limits imposed by the **CompetentAuthority** (*ObtainWWTPEffluentLimits*).

Once the influent is received, the **WWReceiver** determines its destination, either to be sent for treatment or bypass it directly to the river. This decision depends on the (*EvaluateInfluentDestination*) wastewater characteristics (*ObtainInfluentCharacteristics*), *wwtp_j* current treatment efficiency (*ObtainTreatmentEfficiency*) and available capacity (*CalculateAvailableCapacity*). Finally, it also keeps a record of the influent characteristics received (*LogInfluentCharacteristics*) for the **SewageInspector** (as part of the *VerifyDischarge* task) and **WWTreater** (in order to calculate treatment efficiency).

- **WWTreater**: a role responsible for processing the wastewater to reduce its pollutants concentration (*WWTreatment*). Once the treatments ends, treated water is discharged as an effluent to the river. This effluent is analyzed (*ObtainEffluentCharacteristics*) and information is logged so the **CompetentAuthority** can audit it (*VerifyWWTP-Effluent*). Given the effluent and influent characteristics (*ObtainInfluentCharacteristics* / *ObtainEffluentCharacteristics*), **WWTreater** can calculate treatment efficiency (*CalculateTreatmentEfficiency*). This calculation is used to keep *WWTP* as efficient as possible (*AchieveAdequatePerformance*). It is also used to support **WWReceiver** during the discharge price negotiation described before.
- **CompetentAuthority**: this is the role that takes care of ensuring the river has a certain quality level determined by applicable legislation (*EnforceWaterQualityPolicies*). To do so, it should assess river water quality continuously (*AssessRiverWaterQuality*) by means of collecting river sensor data (*CollectRiverSensorData*); these data are requested by entities playing the **WSensor** role. Such entities are expected to be physically located along the river basin. According to the results of this assessment, it may request a **SewageInspector** to verify that every I_i is compliant with what they agreed with *wwtp_j* concerning wastewater discharges (*VerifyDischarge*).

Depending on the current status or even exceptional meteorological conditions, it is possible that the **CompetentAuthority** requires establishing exceptional restrictions on the system (*EstablishExceptionalRestrictions*) to ensure river water quality. The **CompetentAuthority** also uses the above mentioned assessment for two main purposes: First, providing discharge limits for the industry set \mathcal{I} (*CalculateIndustryPermissionLimits*) and effluent limits for the *WWTP* set (*CalculateWWTPEffluentLimits*). In this way, the **CompetentAuthority** can establish pollutant reductions in certain river sections and in general, in the river as a whole. Second, determining reference prices for wastewater discharges (*CalculateDischargeReferencePrices*).

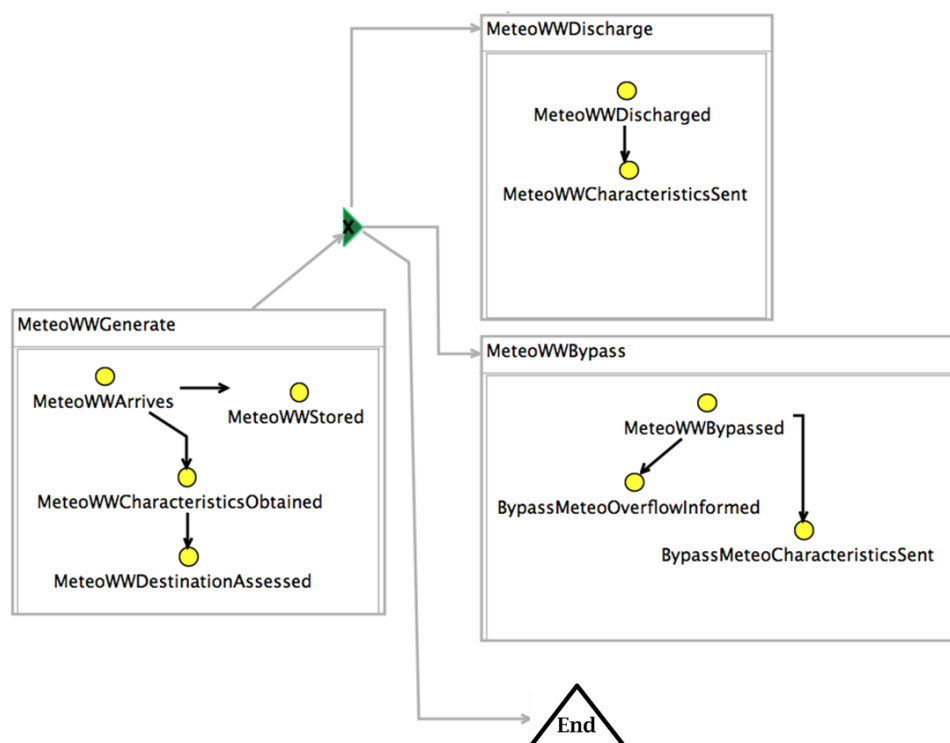
These elements are used by both **IndustrialWWBroker** and **WWReceiver** as part of their negotiation process when discharging industrial wastewater.

Finally, **CompetentAuthority** also checks *WWTP* are compliant with the effluent limits established (*VerifyWWTPEffluent*).

Given that the **CompetentAuthority** has to ensure that the rest of roles are compliant with water quality legislation, all the dependencies that are related to this role are hierarchical ones, since the system requires the **CompetentAuthority** to have a power relation on all the roles to oblige them to coordinate appropriately.

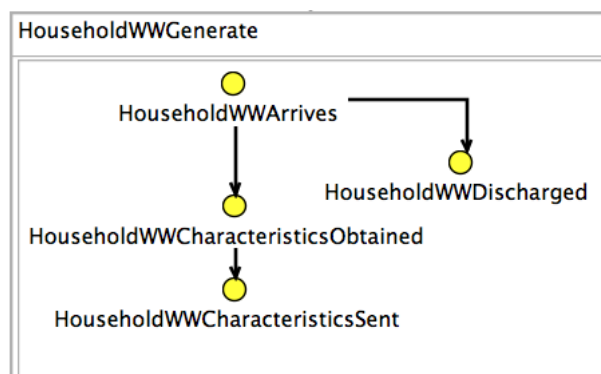
- **SewageInspector**: actors enacting this role are responsible for monitoring activity in the urban sewage system: verifying industrial discharges (*VerifyDischarge*) that are being produced; checking that the influents that are received by the **WWReceiver** (*ObtainLogWWTPInfluents*) that the effluents sent by **IndustrialWWRetainer** (*ObtainLogIndustrialDischarges*) match and that there is no strong deviation that places *WWTP* into a hazardous or unsuitable state (e.g., too many industries discharging more water or with more pollutant concentration than that which was previously agreed, thus affecting *WWTP* processes negatively). The result of this verification is reported to the **CompetentAuthority** to act accordingly.

Appendix B. Landmark Patterns



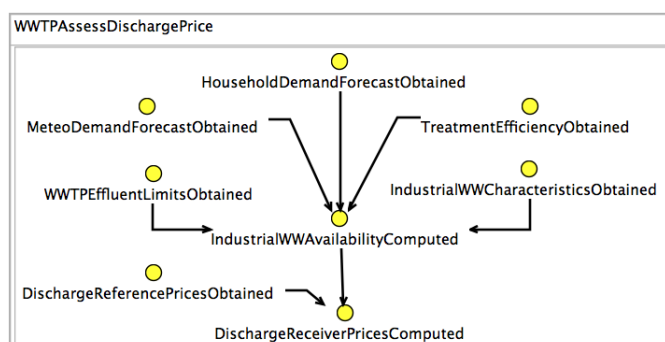
<i>Description</i>	This composite of scenes starts with MeteoWWGenerate , where MeteoRetainer receives wastewater from meteorological phenomena. Wastewater is stored and its characteristics are obtained to decide what to do: store it (End), discharge it to a WWReceiver (MeteoWWDischarge), or inform that a bypass is being performed (MeteoWWBypass).
<i>Players</i>	MeteoWWGenerate : MeteoRetainer , WSensor , CompetentAuthority MeteoWWDischarge : MeteoRetainer , WWReceiver MeteoWWBypass : MeteoRetainer , CompetentAuthority
<i>Results</i>	Wastewater stored, discharged or bypassed. Wastewater characteristics are sent to the WWReceiver or CompetentAuthority .

Figure A1. Landmark patterns for the meteorology discharge. It shows how meteorological wastewater is generated and temporarily stored to be either discharged for normal processing or bypassed to avoid an overflow.



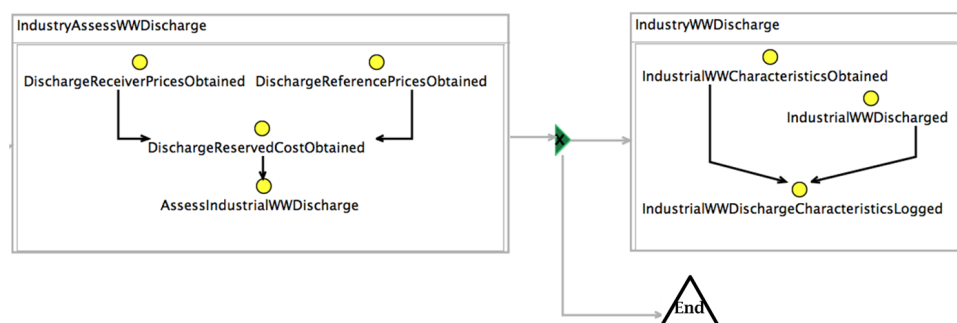
<i>Description</i>	The scene starts with household wastewater being generated. It is managed by a HouseholdWWDischarger role which collects the wastewater from households in a city/town. The role obtains the wastewater characteristics and sends them to a WWReceiver (e.g., the assigned WWTP), and discharges it (not necessarily in that order).
<i>Players</i>	HouseholdWWDischarger, WWSensor, WWReceiver
<i>Results</i>	Wastewater discharged.

Figure A2. Landmark patterns for the household discharge. It shows the flow followed by a household wastewater discharge.



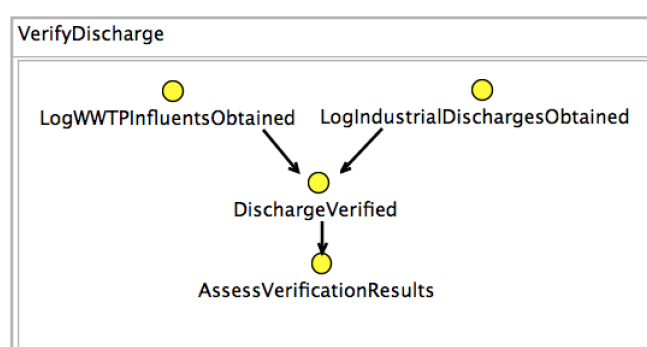
<i>Description</i>	The negotiation started in Figure 4 (in article) and continues in this scene. With all the forecasted discharges, the current efficiency of the treatment, limits on the effluent, the current availability of WWReceiver and the reference price established by the CompetentAuthority , the WWReceiver determines what is the price of accepting a given industrial wastewater mass.
<i>Players</i>	WWReceiver, MeteoWWRetainer, HouseholdWWDischarger, CompetentAuthority, WWTreater, IndustrialWWBroker
<i>Results</i>	Price calculated for accepting a given industrial wastewater mass by WWReceiver .

Figure A3. Landmark patterns for the industry price negotiation. This shows how the price to handle an industrial discharge is determined according to household and meteorological demand forecasts, limitations imposed by norms, volume availability and the wastewater characteristics.



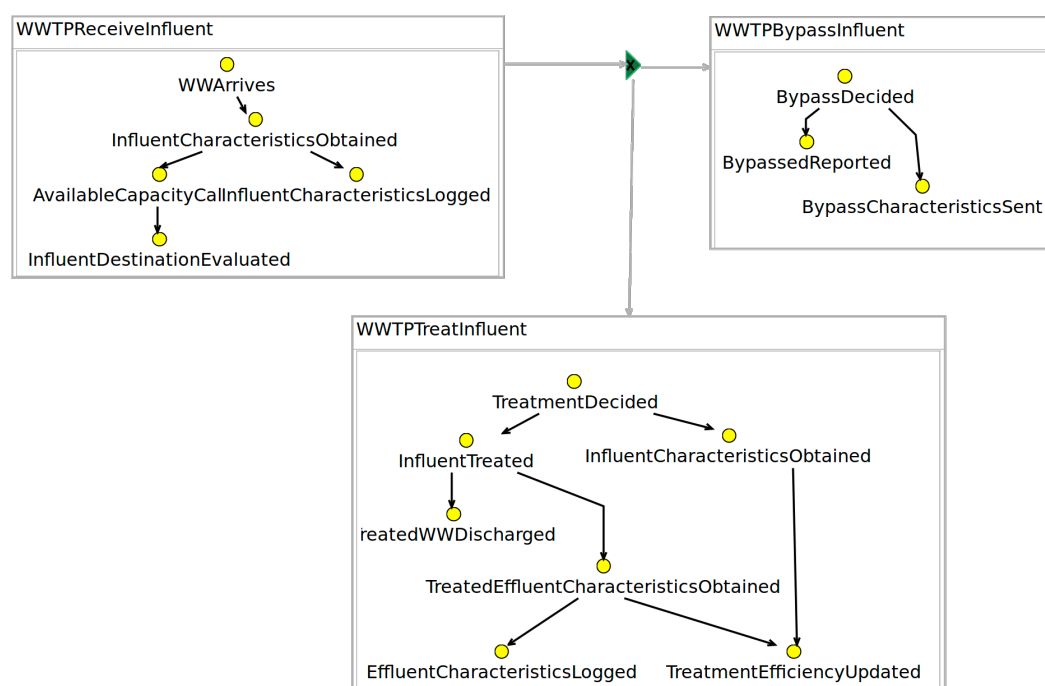
<i>Description</i>	Left scene shows how WWRetainer assesses if it keeps the wastewater it stores or discharges it according to the price obtained by the WWBroker . Right scene occurs when WWRetainer decides to discharge all or part of the wastewater stored.
<i>Players</i>	IndustryAssessWWDDischarge: WWReceiver , WWBroker IndustryWWDDischarge: WWReceiver , WSensor
<i>Results</i>	IndustryAssessWWDDischarge: WWRetainer decides how much wastewater is kept in its storage and how much is discharged. IndustryWWDDischarge: Wastewater mass is discharged and its characteristics are logged. WWRetainer either informs WWReceiver about the agreement or stores the agreement for future verification with logged discharge.

Figure A4. Landmark patterns for the industry discharge. It shows the decision process carried out by the Industry agent to manage its wastewater: keeping it stored or discharging it so a WWTP can process it. Such a decision depends on the processing price and the agent's reserved cost.



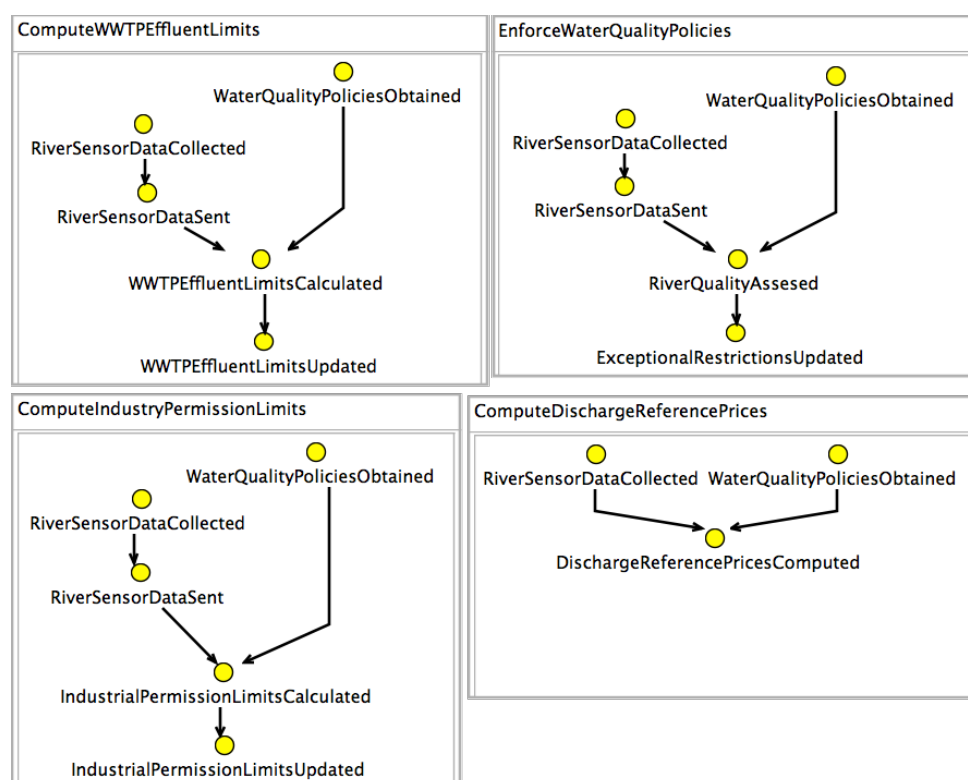
<i>Description</i>	The SewageInspector obtains the industrial discharges logs as well as the masses received by WWReceivers , and checks for any inconsistency that may incur in a sanction.
<i>Players</i>	SewageInspector , IndustrialWWRetainer , WWReceiver
<i>Results</i>	The SewageInspector assesses the results of its verification; this may imply sanctions.

Figure A5. Landmark patterns show how logged information (influent previously received by WWTP) is double-checked against agreed industrial discharges.



<i>Description</i>	WWTPReceiveInfluent: Once a wastewater arrives, its characteristics are obtained and logged. Depending on the available space, it is accepted or bypassed to the river. If the wastewater mass is bypassed, it has to be reported to the CompetentAuthority . If the mass is accepted, the WWTreater logs its characteristics, treats it, logs the treated mass characteristics and updates the treatment efficiency.
<i>Players</i>	WWTPReceiveInfluent: WWReceiver , WSensor WWTPBypassInfluent: WWReceiver , CompetentAuthority , WSensor WWTPTreatInfluent: WWTreater , WSensor
<i>Results</i>	WWTPReceiveInfluent: Wastewater arrives, its characteristics are logged, and it is either accepted or bypassed. WWTPBypassInfluent: Wastewater is discharged to the river and its characteristics are reported to the CompetentAuthority . WWTPTreatInfluent: Wastewater is treated, eventually discharged and its characteristics are logged to update treatment efficiency.

Figure A6. Landmark patterns for WWTP influent treatment. Both diagrams show how wastewater is processed in a WWTP. The left diagram shows the treatment process carried out by WWTP agent: the influent is treated and the effluent characteristics and process efficiency are logged. The right diagram shows how wastewater is bypassed and directly thrown to the river; such a bypass has to be reported to the Competent Authority.



<i>Description</i>	Independent scenes that are mainly performed by a CompetentAuthority to update restrictions, limits and reference prices.
<i>Players</i>	All scenes: CompetentAuthority , WSensor
<i>Results</i>	Each scene updates a specific element that affects the system as a whole: effluent limits for the WWTP, limits for industrial discharges, reference prices for wastewater discharges and raising any exceptional restriction (e.g., heavy rainfall).

Figure A7. Landmark patterns for **CompetentAuthority** control policies. These diagrams show the different carried out by the **CompetentAuthority**: upper-left describes how water quality is assessed and restrictions are enabled to ensure water quality policies. Upper-right show how the effluent limitations for *WWTP* are determined. Lower-left diagram depicts how industrial permits are calculated according to water policies and current river status. Bottom-right diagram shows how discharge references prices are determined to be used later on in the negotiation process between industries and *WWTPs*.

Appendix C. Trace of the Model Simulation

Listing 1. Trace of an execution with two industries and one *WWTP*.

```

***** Tick 1 *****
Industry - 1 - IndustryProduce scene
Industry - 1 - IndustryManageWWTank scene
Industry - 1 requests discharge of 20.0
WWTP - 1 - WWTPAssessDischarge scene
WWTP - 1 requested to receive 20.0 | Admitted: 20.0 (stored: 0.0 | available: 60.0)
Industry - 1 allowed to discharge 20.0, price: 100.0
Industry - 1 - IndustryAssessWWDDischarge scene
Industry - 1 Acceptable price for discharge
Industry - 1 - IndustryWWDDischarge scene
WWTP - 1 - WWTPReceiveInfluent scene
WWTP - 1 receives 20.0 (stored: 20.0 | available: 40.0)
Industry - 2 - IndustryProduce scene
Industry - 2 - IndustryManageWWTank scene
Industry - 2 requests discharge of 20.0

```

```

WWTP - 1 - WWTPAssessDischarge scene
WWTP - 1 requested to receive 20.0 | Admitted: 20.0 (stored: 20.0 | available: 40.0)
Industry - 2 allowed to discharge 20.0, price: 133.34
Industry - 2 - IndustryAssessWWDDischarge scene
Industry - 2 Acceptable price for discharge
Industry - 2 - IndustryWWDDischarge scene
WWTP - 1 - WWTPReceiveInfluent scene
WWTP - 1 receives 20.0 (stored: 40.0 | available: 20.0)
WWTP - 1 - WWTPTreatInfluent scene
WWTP - 1 discharges to river 40.0
WWTP - 1 receives 60.0 (stored: 60.0 | available: 0.0)
***** END Tick 1 *****
***** Tick 2 *****
Industry - 1 - IndustryProduce scene
Industry - 1 - IndustryManageWWTank scene
Industry - 1 requests discharge of 20.0
WWTP - 1 - WWTPAssessDischarge scene
WWTP - 1 requested to receive 20.0 | Admitted: 0.0 (stored: 60.0 | available: 0.0)
Industry - 1 allowed to discharge 0.0, price: 200.0
Industry - 1 - IndustryAssessWWDDischarge scene
Industry - 1 Unacceptable price for discharge
Industry - 2 - IndustryProduce scene
Industry - 2 - IndustryManageWWTank scene
Industry - 2 requests discharge of 20.0
WWTP - 1 - WWTPAssessDischarge scene
WWTP - 1 requested to receive 20.0 | Admitted: 0.0 (stored: 60.0 | available: 0.0)
Industry - 2 allowed to discharge 0.0, price: 200.0
Industry - 2 - IndustryAssessWWDDischarge scene
Industry - 2 Unacceptable price for discharge
WWTP - 1 - WWTPTreatInfluent scene
WWTP - 1 discharges to river 60.0
Industry - 2 overflow, discharges 20.0 to WWTP 1
WWTP - 1 receives 20.0 (stored: 20.0 | available: 40.0)
***** END Tick 2 *****
***** Tick 3 *****
Industry - 1 - IndustryProduce scene
Industry - 1 - IndustryManageWWTank scene
Industry - 1 requests discharge of 40.0
WWTP - 1 - WWTPAssessDischarge scene
WWTP - 1 requested to receive 40.0 | Admitted: 40.0 (stored: 20.0 | available: 40.0)
Industry - 1 allowed to discharge 40.0, price: 133.34
Industry - 1 - IndustryAssessWWDDischarge scene
Industry - 1 Acceptable price for discharge
Industry - 1 - IndustryWWDDischarge scene
WWTP - 1 - WWTPReceiveInfluent scene
WWTP - 1 receives 40.0 (stored: 60.0 | available: 0.0)
Industry - 2 - IndustryProduce scene
Industry - 2 - IndustryManageWWTank scene
Industry - 2 requests discharge of 20.0
WWTP - 1 - WWTPAssessDischarge scene
WWTP - 1 requested to receive 20.0 | Admitted: 0.0 (stored: 60.0 | available: 0.0)
Industry - 2 allowed to discharge 0.0, price: 200.0
Industry - 2 - IndustryAssessWWDDischarge scene
Industry - 2 Unacceptable price for discharge
WWTP - 1 - WWTPTreatInfluent scene
WWTP - 1 discharges to river 60.0
Industry - 2 overflow, discharges 20.0 to WWTP 1
WWTP - 1 receives 20.0 (stored: 20.0 | available: 40.0)
***** END Tick 3 *****
***** Tick 4 *****
Industry - 1 - IndustryProduce scene
Industry - 1 - IndustryManageWWTank scene
Industry - 1 requests discharge of 20.0
WWTP - 1 - WWTPAssessDischarge scene
WWTP - 1 requested to receive 20.0 | Admitted: 20.0 (stored: 20.0 | available: 40.0)
Industry - 1 allowed to discharge 20.0, price: 133.34
Industry - 1 - IndustryAssessWWDDischarge scene
Industry - 1 Acceptable price for discharge
Industry - 1 - IndustryWWDDischarge scene
WWTP - 1 - WWTPReceiveInfluent scene
WWTP - 1 receives 20.0 (stored: 40.0 | available: 20.0)
Industry - 2 - IndustryProduce scene
Industry - 2 - IndustryManageWWTank scene
Industry - 2 requests discharge of 20.0
WWTP - 1 - WWTPAssessDischarge scene
WWTP - 1 requested to receive 20.0 | Admitted: 20.0 (stored: 40.0 | available: 20.0)
Industry - 2 allowed to discharge 20.0, price: 166.67
Industry - 2 - IndustryAssessWWDDischarge scene
Industry - 2 Acceptable price for discharge
Industry - 2 - IndustryWWDDischarge scene
WWTP - 1 - WWTPReceiveInfluent scene

```

```

WWTP - 1 receives 20.0 (stored: 60.0 | available: 0.0)
WWTP - 1 - WWTPtreatInfluent scene
WWTP - 1 discharges to river 60.0
***** END Tick 4 *****

```

References

1. Rendón-Sallard, T.; Sánchez-Marrè, M.; Aulinas, M.; Comas, J. Designing a Multi-Agent system to simulate scenarios for decision-making in river basin systems. In Proceedings of the Ninth International Conference of the Catalan Association of Artificial Intelligence (CCIA'2006), Perpignan, France, 26–27 October 2006; Frontiers in Artificial Intelligence and Applications Series; Artificial Intelligence Research and Development; IOS Press: Amsterdam, The Netherlands, 2006; Volume 146, pp. 291–298. ISBN 1-58603-663-7.
2. Poch, M.; Comas, J.; Rodríguez-Roda, I.; Sánchez-Marrè, M.; Cortés, U. Designing and building real environmental decision support systems. *Environ. Model. Softw.* **2004**, *19*, 857–873. [\[CrossRef\]](#)
3. Cortés, U.; Sánchez-Marrè, M.; Ceccaroni, L.; R-Roda, I.; Poch, M. Artificial intelligence and environmental decision support systems. *Appl. Intell.* **2000**, *13*, 77–91. [\[CrossRef\]](#)
4. Comas, J.; Poch, M.; Rodríguez-Roda, I.; Cortés, U. *Eleven Years of Experience in Designing and Building Real environmental Decision Support Systems: What Have We Learnt?* Universitat de Girona: Catalonia, Spain, 2002.
5. Benedetti, L.; Prat, P.; Nopens, I.; Poch, M.; Turon, C.; De Baets, B.; Comas, J. A new rule generation method to develop a decision support system for integrated management at river basin scale. *Water Sci. Technol.* **2009**, *60*, 2035–2040. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Wooldridge, M. Agent-based software engineering. *IEEE Proc. Softw.* **1997**, *144*, 26–37. [\[CrossRef\]](#)
7. Cortés, U.; Poch, M. *Advanced Agent-Based Environmental Management Systems*; Birkhäuser Basel AG: Basel, Switzerland, 2009.
8. Wooldridge, M.; Jennings, N.R.; Kinny, D. The GAIA methodology for agent-oriented analysis and design. *Auton. Agents Multi Agent Syst.* **2000**, *3*, 285–312. [\[CrossRef\]](#)
9. Zambonelli, F.; Jennings, N.R.; Wooldridge, M. Developing multiagent systems: The Gaia methodology. *ACM Trans. Softw. Eng. Methodol. (TOSEM)* **2003**, *12*, 317–370. [\[CrossRef\]](#)
10. DeLoach, S.A.; Wood, M.F.; Sparkman, C.H. Multiagent systems engineering. *Int. J. Softw. Eng. Knowl. Eng.* **2001**, *11*, 231–258. [\[CrossRef\]](#)
11. Wood, M.F.; DeLoach, S.A. An overview of the multiagent systems engineering methodology. In *International Workshop on Agent-Oriented Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 207–221.
12. Bresciani, P.; Perini, A.; Giorgini, P.; Giunchiglia, F.; Mylopoulos, J. Tropos: An agent-oriented software development methodology. *Auton. Agents Multi Agent Syst.* **2004**, *8*, 203–236. [\[CrossRef\]](#)
13. Padgham, L.; Winikoff, M. Prometheus: A methodology for developing intelligent agents. In *International Workshop on Agent-Oriented Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 174–185.
14. Sturm, A.; Shehory, O. A comparative evaluation of agent-oriented methodologies. In *Methodologies and Software Engineering for Agent Systems*; Springer: Boston, MA, USA, 2004; pp. 127–149.
15. Zambonelli, F.; Omicini, A. Challenges and research directions in agent-oriented software engineering. *Auton. Agents Multi Agent Syst.* **2004**, *9*, 253–283. [\[CrossRef\]](#)
16. Vázquez-Salceda, J.; Vasconcelos, W.W.; Padget, J.; Dignum, F.; Clarke, S.; Roig, M. ALIVE: An agent-based framework for dynamic and robust service-oriented applications. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: International Foundation for Autonomous Agents and Multiagent Systems, Toronto, ON, Canada, 10–14 May 2010; Volume 1, pp. 1637–1638.
17. Álvarez Napagao, S.; Cliffe, O.; Vázquez Salceda, J.; Padget, J. Norms, organisations and semantic web services: The ALIVE approach. In Proceedings of the Second Multi-Agent Logics, Languages, and Organisations Federated Workshops, Turin, Italy, 7–10 September 2009; pp. 1–2.
18. Dignum, F.; Dignum, V.; Padget, J.; Vázquez-Salceda, J. Organizing web services to develop dynamic, flexible, distributed systems. In Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services, Kuala Lumpur, Malaysia, 14–16 December 2009; ACM: New York, NY, USA, 2009; pp. 225–234.
19. Gómez-Sebastià, I.; Oliva-Felipe, L.; Cortés, U.; Verdaguer, M.; Poch, M.; Rodríguez-Roda, I.; Vázquez-Salceda, J. A norm-aware multi-agent system for social simulations in a river basin. In *Intelligence Systems in Environmental Management: Theory and Applications*; Springer International Publishing: Cham, Switzerland, 2017; pp. 67–90.
20. Aldewereld, H.; Dignum, V. Operetta: Organization-oriented development environment. In *Languages, Methodologies, and Development Tools for Multi-Agent Systems*; Springer: Cham, Switzerland, 2011; pp. 1–18.
21. Aldewereld, H.; Álvarez-Napagao, S.; Dignum, V.; Jiang, J.; Vasconcelos, W.; Vázquez-Salceda, J. OperA/ALIVE/Operetta. In *Social Coordination Frameworks for Social Technical Systems*; Springer International Publishing: Cham, Switzerland, 2016; pp. 173–196.
22. Verdaguer, M.; Clara, N.; Poch, M. Ant colony optimization-based method for managing industrial influents in wastewater systems. *AIChE J.* **2012**, *58*, 3070–3079. [\[CrossRef\]](#)

-
23. North, M.J.; Collier, N.T.; Ozik, J.; Tatara, E.R.; Macal, C.M.; Bragen, M.; Sydelko, P. Complex adaptive systems modeling with Repast Symphony. *Complex Adapt. Syst. Model.* **2013**, *1*, 3. [[CrossRef](#)]
 24. Gómez-Sebastià, I. NoMoDEI: A framework for Norm Monitoring on Dynamic Electronic Institutions. Ph.D. Dissertation, Universitat Politècnica de Catalunya, Barcelona, Spain, 2016.