# Semantic Mediation in Smart Water Networks

Georgios M. Milis, Demetrios G. Eliades, Christos G. Panayiotou, and Marios M. Polycarpou
KIOS Research Center for Intelligent Systems and Networks
Department of Electrical and Computer Engineering
University of Cyprus
Nicosia, Cyprus
Email: {milis.georgios, eldemet, christosp, mpolycar}@ucy.ac.cy

***Abstract*—Water Distribution Networks (WDN) are the infrastructures responsible for delivering drinking water to consumers. The effective monitoring and control of these systems is of vital importance since malfunction may significantly affect the health, safety, security and/or economic well-being of people. The advancements in coupling WDN with the ICT infrastructure, combined with the more recent introduction of smart sensing and actuation technologies, have enabled the enhancement of "Supervisory Control And Data Acquisition (SCADA)"-based applications. These applications in current water systems assume pre-defined configuration and characteristics of the involved components (sensors, actuators, controllers, etc.). This work explores how semantic mediation techniques may contribute to the online configuration of the monitoring and control architectures by exploiting and reasoning over the capabilities of deployed devices.**

## I. Introduction

Water Distribution Networks (WDN) are considered among the most Critical Infrastructures, along with Power and Telecommunications Systems, since they can significantly affect the health, safety, security and/or economic well-being of the citizens when disrupted or when their operation degrades. It is therefore of vital importance to monitor and control these systems in a way to minimize disturbances from the normal operation.

The objectives of WDNs are to deliver water of sufficient quality and quantity to the consumers, maximize the efficiency of this delivery, as well as guarantee the safety of the system. Essentially, WDNs are large-scale systems, which consist of pipe networks and dynamical elements such as water storage tanks, pumps and valves to control pressures and flows in the system, as well as sensors measuring various hydraulic and quality water characteristics. In practice, the state-of-art in the monitoring and control of these systems, involves the use of Supervisory Control And Data Acquisition (SCADA) systems coupled with an ICT infrastructure that enables the transfer of data and further processing of sensing and actuation signals [1]. Figure 1, presents the typical architecture of a WDN monitoring and control system, capturing both the hydraulic and quality characteristics. Hydraulic sensors measure tank water levels, hydraulic heads of junctions (i.e., the surface elevation of the junction comparing to some reference level), flows and pressures, while quality sensors measure pH, chlorine concentrations, Oxidation Reduction Potential, Total Organic Carbon, etc. The inputs to the system are generated by hydraulic actuators (e.g., valves, pumps), as well as quality actuators (e.g., chlorine disinfection boosters). The control decisions are implemented based on pre-defined rules or control algorithms that map the measurements to appropriate actions. It is also noted that, in practice, water utilities typically employ manual sampling and controlling. On the other hand, utilities in some countries have started employing various types of sensors, such as Automatic Meter Readers (AMR) to measure water consumption in real-time, as well as other hydraulic and quality sensors; in contrast, real-time monitoring and control algorithms are still under research and have not been exploited by the utilities.

As the number of sensors and actuators in the WDNs increases, so does the complexity in managing these elements and reconfiguring the system whenever a sensor or actuator is added or removed. In practice, the measurements from the WDN are retrieved by physical devices of appropriate types and of a variety of (vendor-dependent) specifications. Therefore, either the control engineer needs to design the control law based on pre-acquired knowledge about the available devices and their specifications or the technicians that install the devices need to know the specifications of the SCADA system and the control algorithms in advance, so as to install appropriate devices. These cases show the existing inflexibility of the current WDN monitoring and control architectures. The recent advancements in Smart Water Networks and related sensing and actuation capabilities create additional considerable opportunities and challenges towards offering flexibility in the monitoring, control and event detection architectures [2]–[4]. However, in time, changes in the sensing and actuation capabilities, may necessitate changes in the monitoring and control algorithms, as well, which implies the need for intervention of human experts (e.g., control engineers to update the controllers given any new sensing and actuation capabilities). This is where the main motivation of this work stems from.

The present work contributes to accommodating component changes online, by offering flexibility with the introduction of a monitoring and control architecture that allows online configurability. The proposed architecture is enriched with a semantic mediation layer, which stores structured knowledge about the available components (e.g., measuring and actuation devices and control implementations), and a semantic mediation agent $\Sigma$, which performs semantic reasoning after any change in the available components and takes decisions about the online re-configuration of the system. The situation awareness is achieved by utilizing a pre-designed domain ontology that fully describes the types of the components, their characteristics, their locations, as well as the physical properties they measure or act upon.

The key advantage of the proposed architecture is the in-

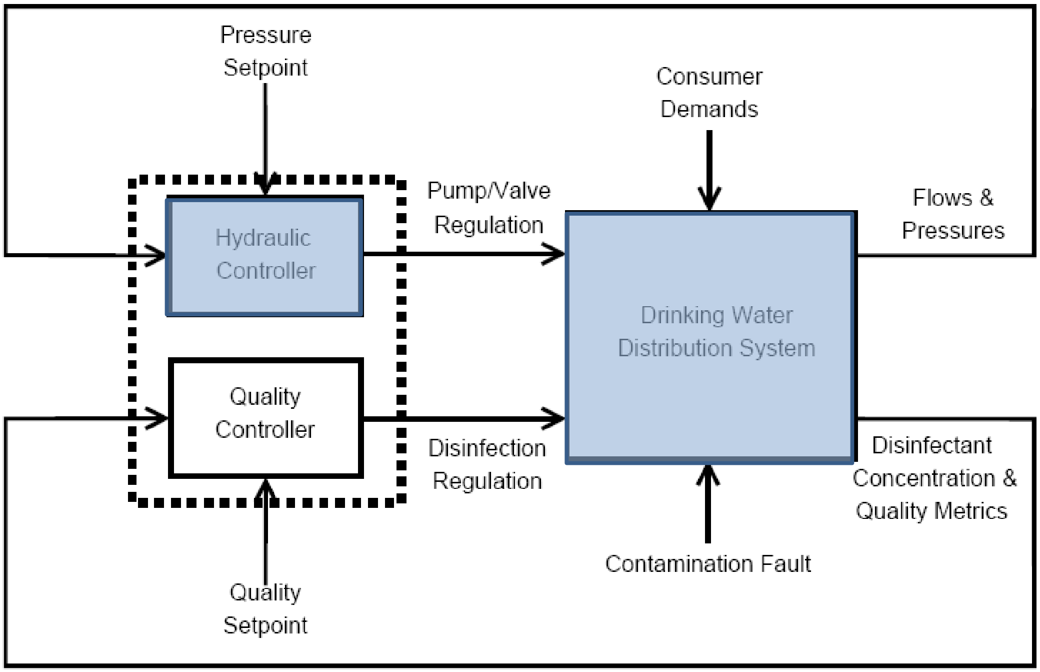


Fig. 1: The typical architecture of a WDN Monitoring and Control system. This work focuses on the components highlighted with blue transparent color.

herited flexibility and automation of the components' selection, through the semantic layer. Structuring the knowledge representation, enables a machine (i.e., the agent $\Sigma$) to undertake the tasks that would otherwise be undertaken by humans. On the other hand, the main drawback of the architecture is that it requires spending considerable effort offline, to create correct knowledge models and semantic annotations for the involved components. Although this effort is only required to be allocated once or in rather rare cases, it is a trade-off that needs to be considered depending on the application and the estimated benefits.

The paper is organized as follows: Section II provides some background information in relation to water distribution systems, Section III formulates the problem, Section IV presents the proposed architecture and methodology, followed by Section V where the knowledge model for semantic mediation is presented. Finally, Section VII uses illustrative cases to discuss the semantic reasoning mechanism for the online components' composition and Section VIII concludes the paper providing also some insight to future directions.

## II. Background on modeling and control of hydraulics

Modeling methodologies for hydraulic and quality dynamics and their faults have received significant attention during the last decade [5], [6], and are still an area of active research. The present work focuses on the hydraulic characteristics of a WDN, so as to achieve a simple presentation of the proposed semantic enhancements and demonstrate the concept.

The hydraulic feedback control problem in water systems can be defined as the problem of computing at each discrete time $k$, the input vector $u(k)$, representing the instructions to be given to the pumps and valves, so that the measured hydraulic parameters $y(k)$ (tank water levels, hydraulic heads, as well as the pipe flows) operate within certain bounds or follow a reference signal vector $r(k)$ specified for safe operation. The control law is given by the $f_u(\cdot)$ function given in (1).

$$u(k) = f_u(y(k), r(k)). \tag{1}$$

The above control application depends on the measurement or estimation of the hydraulic parameters, defined as the hydraulic analysis problem in WDNs; that is, compute the hydraulic head at each junction (i.e., surface elevation comparing to a reference level), the water levels at each tank and the flows at each pipe. To solve this problem, the topology of the network and pipe characteristics, the control inputs, as well as the demand at each node, are assumed known *a priori*. In general, structural information of the network is available by the water utilities, while pipe characteristics may require field measurements and nodal demands at each discrete time can only be estimated using historical data and other hydraulic measurements available (if no online demand sensors are used by the utility to monitor each consumer).

The dynamic relation of water flow in pipes and the differences in the hydraulic heads can be described by a set of ordinary differential equations. In practice, however, the heads and flows are approximated using an iterative optimization algorithm (e.g. gradient descent), in discrete time and in steady state, so that the conservation of mass and energy is satisfied [7]. For example, consider a water distribution network composed of pipes, junctions and water storage units. The topology of this network can be represented as a graph with edges corresponding to pipes, and nodes corresponding to junctions and water storage units. At discrete time $k$ with sampling time $\Delta t$, let $d_i(k)$ be the consumer demand outflow at the $i$-th junction node, and let $w_j(k)$ correspond to the flow in the $j$-th pipe connected to junction $i$ ($j \in \mathcal{A}_i$ where $\mathcal{A}_i$ is the set of pipe indices which are connected to the $i$-th node, assuming that inflows have a positive sign and outflows have a negative sign). In accordance to the principle of mass conservation, the sum of all pipe inflows and pipe outflows must equal to the demand (Kirchhoff's junction rule), such that:

$$\sum_{j \in \mathcal{A}_i} w_j(k) = d_i(k) \tag{2}$$

Furthermore, in accordance to the principle of energy conservation, the flow-headloss relationship across each link in the network must be balanced. Let $h_i(k)$ be the *hydraulic head*, i.e. a measurement of water pressure expressed in length units, at the $i$-th node. For water moving from node $j$ (higher head) to node $i$ (lower head) with flow $w_l(k)$ in the $l$-th pipe, the flow-headloss relationship is given by:

$$h_j(k) - h_i(k) = f_h(w_l(k)) \tag{3}$$

where $f_h(\cdot)$ is a nonlinear function, such that $f_h(w_l(k)) = \alpha_r w_l(k)^{\alpha_f} + \alpha_m w_l(k)^2$, which depends on the pipe resistance coefficient $\alpha_r$, the flow exponent $\alpha_f$ and the minor loss coefficient $\alpha_m$. These parameters are computed using empirical methods [8]. Therefore, for a water distribution network, the set of hydraulic equations is constructed, and at each discrete time, a gradient optimization algorithm is solved to estimate the heads at each junction/tank, using the current demand flows, current control inputs and current tank heads [7].

Tanks are dynamic elements in the system and can be considered as nodes in the water distribution network; the head state of the $i$-th water tank node is given by (4).

$$h_i(k+1) = h_i(k) + \frac{\sum\limits_{j \in \mathcal{A}_i} w_j(k)}{f_{T_i}(h_i(k))} \Delta t, \tag{4}$$

where the tank head $h_i(k)$ corresponds to the relative tank water level plus the tank elevation, and the function $f_{T_i}(\cdot)$ computes the cross-sectional area of the $i$-th tank at a certain height. Initial tank heads are typically known.

Currently, a number of off-the-shelf software tools are used to perform the hydraulic analysis in water distribution networks, such as the open-source EPANET [8].

## III. Challenge Formulation

Consider the network in Fig. 2. The arrows indicate the flow of water in pipes, as well as the inflow of the tank ($w_i$, $i = 0, ..., 9$). The tank flow is indicated by $w_0$. The nodes

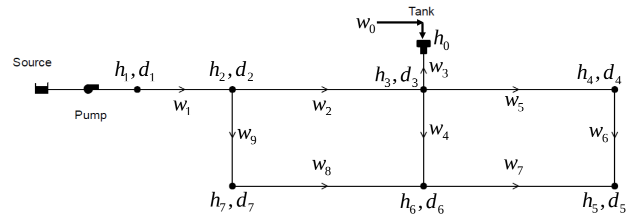


Fig. 2: A simple WDN with six junction nodes; water is supplied by a reservoir and a tank. When the tank water level goes below 110ft, the pump is activated, and when the tank water level goes above 140ft, the pump stops.

indicate the junctions with their hydraulic heads represented by $h_i,\ i = 0, ..., 7$ and the corresponding consumer demand outflows $d_i$. The tank head is indicated by $h_0$.

Consider also the control objective to regulate the tank head $h_0(k)$ at a given reference level $r_0(k)$ (or within a certain bound), at discrete times $k$, through the application of a control law as in (1). In this example, the pump is activated only when the tank water levels has reached a minimum value (110 ft), and the pump stops working when the water level has reached a maximum value (140 ft).

As explained in Section II, the tank and junction heads, if not directly measured, are estimated iteratively at each discrete time $k$, using the mass conservation equations on junctions, as well as the energy conservation equations that require the headloss functions. The parameters of the headloss functions $f_h(\cdot)$ are considered known or are computed empirically. In addition, the demand flows at each consumption node $d_i,\ i = 1, ..., 7$, as well as the initial tank head $h_0(0)$, are known. At time $k$, an optimization algorithm is used to compute the unknown states. Then the new tank head is computed again and the problem is solved for the next discrete time $k + 1$.

The tank head regulation process can be illustrated with the block diagram of Fig. 3, where the vector of the states of the WDN (i.e., the tank and junction heads and pipe flows) is defined as $x \in \mathcal{R}^n$, the measurements' vector produced by the installed set of measurement devices ($\mathcal{S}$) is given by $y \in \mathcal{R}^p$, while $\hat{x} \in \mathcal{R}^n$ denotes the vector of the estimated system states after the application of the iterative optimization algorithms. The diagram shows that the controller $K$ uses the measured or estimated tank head as input and produces action signal that is then utilized by the set of actuators $\mathcal{A}$ (e.g., pumps, valves) to act on the system and affect the tank head.

The control implementation for the regulation of the tank head is typically comprised of a set of pre-defined rules, such as the following: `IF <tank-head> <expression referring to tank head> THEN <action to be performed on the specific tank>`. In current practice, the design of such control architectures in WDNs, is based on a fixed configuration of specific sub-components, with specific measurement and actuation devices deployed and predetermined control laws. That is, the utilities need to decide in advance the types of components to use and where in the network's topology, as well as implement in advance the respective control rules. Moreover, integration of control components (e.g., after a change happens) typically requires manual configuration by an experienced engineer. It is therefore typical to interrupt the normal system operation to modify the control system, i.e., to manually configure the new sensor/actuator in the SCADA system, as well as manually modify the logic in the micro-controller (for when to turn-on/turn-off a pump).

As the scale of the system increases, however, it becomes a real challenge in having highly specialized personnel which is able to perform these increasingly complex functions. A "smart water" cyber-physical implementation would be expected to be able to adapt to changes in the composition of the control system (e.g., removal of a faulty tank water-level sensor, addition of pipe flow or tank inflow measurement, updated set of control rules, etc.) and perform any necessary re-configurations online, in order to avoid the need of downtime.

A solution to this problem would be the introduction of an intermediate layer, which would be aware of any new situation, have at its disposal a set of components with their own characteristics and input/output mappings structurally described and be capable of taking informed online decisions on how to implement the control system. Such mediation architectures have been proposed for the automatic composition of web services by the Internet community. The composition is achieved using standard ontology frameworks that allow semantic composition of services/components, such as *OWL-S* [9] or frameworks for semantic annotation of *RESTfull* services [10]. Recent efforts in the framework of the *Internet of Things* paradigm, promote the use of these technologies in a cyber-physical perspective, addressing the additional spatio-temporal challenges involved with the interaction with the physical world. Therefore, ontologies have been proposed, dedicated to sensor annotations such as the *SSN* [11] and the *SensorML* standard [12]. Individual components are then semantically described using the pre-defined semantic description frameworks in combination with domain ontologies, which allows the selection of the appropriate components in a composition aiming to achieve a more complex objective.

The following section presents the proposed architecture that offers a promising solution to the above described challenge.

## IV. Methodology and Proposed Architecture

The proposed architecture for the tank head regulation problem, is depicted in Fig. 4, where $I \in \{0, 1, 2, ...\}$ is the index of the control architecture configuration. In configuration $I$, the operation of the WDN is monitored by a set of sensors $\mathcal{S}^{(I)}$, e.g., tank and junction head, as well as pipe flow sensors. The measurements may pass through a set of functions $\mathcal{F}^{(I)}$ (such function might also be the state estimator that computes missing state-values). Then the appropriate control implementation $K^{(I)}$ is chosen to drive the set of actuators $\mathcal{A}^{(I)}$.

In order to achieve the objective of shifting from configuration $I$ to $I+1$ when required, the use of a *Semantic Mediation Agent* $\Sigma$ is proposed, as introduced earlier, which is responsible for taking the decisions and configuring the control structure. The agent $\Sigma$ first detects and identifies any new component(s) added. The physical communication among components is facilitated through an assumed existing

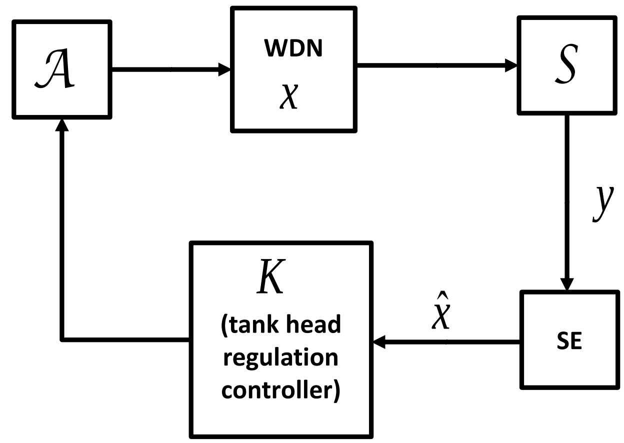


Fig. 3: Block diagram of the tank head regulation implementation. A set of sensors $\mathcal{S}$ measure part of the states of a WDN (i.e., the tank and junction heads and pipe flows), defined by the vector $x \in \mathcal{R}^n$, producing the vector $y \in \mathcal{R}^p$. Then a state estimator (SE) is used, estimating the complete set of states, by producing the vector $\hat{x} \in \mathcal{R}^n$, which is then fed to the controller $K$ to help it compute the input to the set of actuators $\mathcal{A}$ (e.g., pumps, valves).

communication protocol, e.g., with extensions to the currently adopted SCADA systems (the details of this fall outside the scope of this work). Subsequently, $\Sigma$ becomes aware of the characteristics and capabilities of the new components, which is exactly the emphasis of this work. The use of *ontological knowledge models* and *semantic annotation and mediation* techniques are proposed [13], [14]. Each component is assumed pre-annotated with certain "tags" that will describe its characteristics and capabilities. Once this information is received by $\Sigma$ and is stored in its knowledge model $\Lambda$, it is integrated with the existing knowledge available about the overall system. The whole of the information is in turn used to potentially infer new (implicit) knowledge, denoted by $\Lambda_m$, that is subsequently utilized to reconfigure the existing control architecture considering all available components, e.g., to utilize the state estimation function if the tank water level (or tank head) is not measured directly. Every time a new measurement (e.g., a flow sensor) or actuation unit (e.g., a valve) is added and/or a controller implementation is added, agent $\Sigma$ detects and identifies it. Its main task is to become aware of the new component's functionality, properties and characteristics (through semantic annotations) and add it to its knowledge-base and/or obtain additional information from other sources, such as from an internet web-service or from human users. It is emphasized here that any shift from configuration $I$ to $I+1$ happens strictly in the time between subsequent discrete time steps $k$ and no interruption of a running process is performed. The time constants of the WDN allow the appropriate selection of the time between consecutive control steps, so as to be enough for $\Sigma$ to complete the reasoning and decide the new configuration.

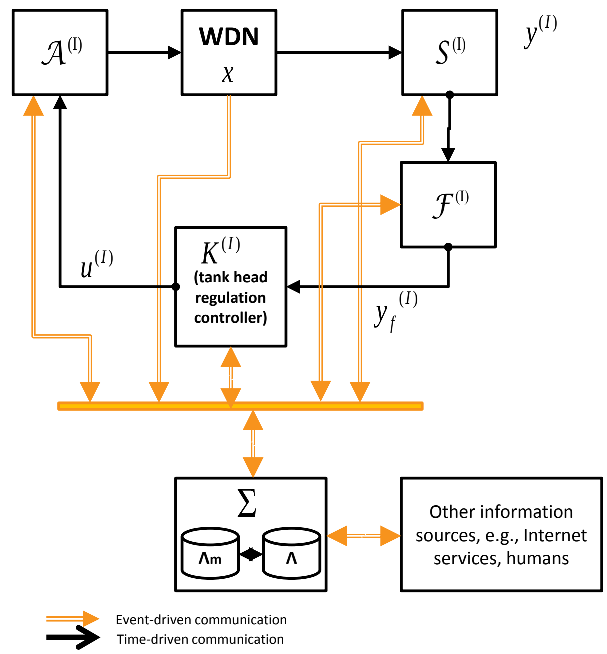

Fig. 4: Block diagram of the proposed tank head regulation architecture implementation, where the agent $\Sigma$ is introduced with event-driven communication and knowledge exchange capabilities, so as to reason about appropriate wiring of components. The diagram shows the set of used sensors $\mathcal{S}^{(I)}$, the utilized controller $K^{(I)}$, the set of transformation functions $\mathcal{F}^{(I)}$, as well as the set of actuators $\mathcal{A}^{(I)}$. The time-driven communication concerns vectors of signals each time with appropriate dimension. The event-driven communication comprises two-way cyber communication with components, to facilitate their semantic annotations' sharing, as well as the online wiring. Note that communication with the WDN is one-way since the system is assumed as not communicating anything directly to the agent $\Sigma$ (any knowledge about it comes from human or cyber sources).

## V. Semantic Layer Modeling

## VI. Semantic Mediation

In general, knowledge models help in facing the inter-operation issues by implementing structured representations of domain knowledge and by providing suitable reasoning facilities to make best possible use of the combined stored knowledge. An illustrative scenario follows, aiming at clarifying how the knowledge model is built, as well as, how the logical (semantic) reasoning is performed over the stored knowledge facts to implement the decision mechanism for the online configuration of the WDN control system.

Let assume the simple WDN of Fig. 2, where the initial configuration ($I = 0$) of the tank head regulation system consists of a sensor measuring the tank head $h_0$ (or water level) in meters ($m$) and an on/off pump which increases the pressure in another place of the system, in order to for water to flow into the tank. In addition, a simple programmable logic controller (PLC) is considered available, implementing "IF-THEN" rules that, given the desired level of water in the tank, map the measured level to decisions on whether to close or open the pump. The introduced system contains a number of "things". There are essentially one sensor ($s_1$), the physical property "tank-head" ($q_1$), the measurement unit "meters" ($m_1$), as well as the "tank" and the "pump-position" as system locations ($l_1$ and $l_2$ respectively). These "things" represent either cyber-physical components or other types of linguistic terms which are modeled as knowledge objects (elements of a set $\mathcal{N}$) in the dedicated ontology.

Moreover, the type-set of each object (e.g., "meters" is a measurement unit while "tank" is a location) is also defined in the ontology. Therefore, the set of sensors ($\mathcal{S}$), the set of locations ($\mathcal{L}$), the set of physical properties ($\mathcal{Q}$), as well as the set of measurement units ($\mathcal{M}$) are defined. In order to facilitate the online configuration of the control system, with composition of appropriate cyber and/or physical components, these are modeled as abstractions of inputs and outputs. Therefore, the sets $\mathcal{I}$ and $\mathcal{O}$ represent the inputs and outputs respectively, assuming also that the inputs and outputs properly inherit the semantic characteristics of the components. These are essentially elements of a types-set $\Omega$ and subsets of $\mathcal{N}$. Each of these type-sets has a separate finite cardinality $n_V$, where $V$ is any element of $\Omega$.

In addition, the full understanding of the meaning is facilitated by the relations between the existing objects. That is, "meters" ($m_1 \in \mathcal{M}$) is a measurement unit of "tank-head" ($q_1 \in \mathcal{Q}$), while the sensor $s_1 \in \mathcal{S}$ is located on the "tank" ($l_1 \in \mathcal{L}$). These relations can be modeled as (non-balanced) bipartite graphs where edges define mappings among vertices of a domain object-set to a range object-set. The definition is given below:

*Definition:* $G(V^o, V^d, E^{(V^o,V^d)})$: defines a non-balanced bipartite graph (called here also relation graph) with vertices being the elements of the sets $V^o = \{v_i^o | i = 1, 2, ..., n_{V^o}\}$ and $V^d = \{v_j^d | j = 1, 2, ..., n_{V^d}\}$, $V^o, V^d \in \Omega$, and edges being the elements of the set $E^{(V^o,V^d)} = \{(v_i^o, v_j^d) | v_i^o \in V^o, v_j^d \in V^d\}$, which represent the connections between elements of the origin set $V^o$ to elements of the destination set $V^d$.

Examples of relations might be the relation between sensor and locations $G(\mathcal{S}, \mathcal{L}, E^{(\mathcal{S},\mathcal{L})})$, the relation between sensors and measurement units $G(\mathcal{S}, \mathcal{M}, E^{\mathcal{S},\mathcal{M}})$, the relation between measurement units and physical properties $G(\mathcal{M}, \mathcal{Q}, E^{(\mathcal{M},\mathcal{Q})})$, as well as two relations between location objects representing the case where a location "is part of" another location $G(\mathcal{L}, \mathcal{L}, E_1^{(\mathcal{L},\mathcal{L})})$ and the case where a location "is adjacent to" another location $G(\mathcal{L}, \mathcal{L}, E_2^{(\mathcal{L},\mathcal{L})})$. Note the use of the subscript on the set of edges $E$, which helps differentiating between multiple relations over same pair of sets. Graphical representations follow later in the paper to clarify the meaning of relations.

The above ontology-based model in fact comprises the convention/agreement between all physical and soft/cyber components that interact in the control system implementation, about the interpretation of their capabilities (e.g., inputs/outputs) and the domain in which they operate. The description of a new object using the pre-defined ontology is called "semantic annotation of the object". For instance, the semantic annotation of the sensors in this work comprises information about the location of the corresponding device and the units of the produced values (e.g., meters). Then, the pre-defined ontology is exploited by the $\Sigma$ component to take rational decisions about which sensors, actuators and control implementations to use and how to organize these components in the composition of the control system architecture in order to be able to satisfy the control objectives. The architecture allows also the use of other functions to process the measurements before feeding them to the controller.

The knowledge model adopted in this work and described above, is implemented as a lightweight ontology to facilitate the presentation of the concept. This ontology will be merged in the future with existing standard ontology frameworks that allow semantic composition of services/components, as has been discussed in Section II. The logical decisions are taken based on inference rules, written using the SPARQL Protocol and the RDF Query Language, [15]. The use of ontologies as a standard mean for structuring and representing the knowledge, enables the accessibility and utilization of this knowledge by machines (i.e., the agent $\Sigma$ in this work). Moreover, it enables the management of the knowledge separately from the application. Without a standard representation of the knowledge, agent $\Sigma$ would either not be able to reason about components' capabilities or, if such information was hard-coded in its implementation, continuous human intervention would have been unavoidable.

### A. Semantic annotations and compositions of components

The semantic annotation of the components in fact achieves the encoding of the required knowledge in machine readable format, to facilitate the automatic reasoning for the online composition of the control system. In order to facilitate understanding, the semantic annotation of a sensor $s_i,\ i = 1,...,n_{\mathcal{S}}$ is graphically introduced as an example, in Fig. 5. The sensor has one input $t_i,\ i = 1,...,n_{\mathcal{T}}$ and one output $o_i,\ i = 1,...,n_{\mathcal{O}}$. The components (cyber and physical) are shown in the "Implementation Layer", while they are also represented by knowledge objects in the "Knowledge Model Layer". The knowledge objects are shown with circles and the type-sets are shown with dashed-line rectangle containers. The index $j$ in each of the type-sets is defined in the same way as $i$, however, $j \neq i$. The relations between the objects are shown by the edges (thick-black or thin-grey depending on whether the edge is part of the object's semantic annotation or not). For further clarity, the knowledge model is split into four layers, separated by dark blue dashed lines: i) the "System Components" layer which contains the knowledge objects that represent the actual implementations of components, ii) the "Inputs/Outputs" layer which contains all inputs and outputs of components, iii) the "Thematic Knowledge" layer which contains all other domain specific knowledge objects that are used to annotate the components, and iv) the "Functions" layer which hosts knowledge objects representing the signal processing functions.

It can be seen that the input of the sensor is annotated as representing a physical property signal ($q_i,\ i = 1,...,n_{\mathcal{Q}}$) from a specific location ($l_i,\ i = 1,...,n_{\mathcal{L}}$). On the other hand, the output is associated with some measurement unit at some location (e.g., tank of junction) of the WDN. Additional inputs and outputs can be modeled in the same way.

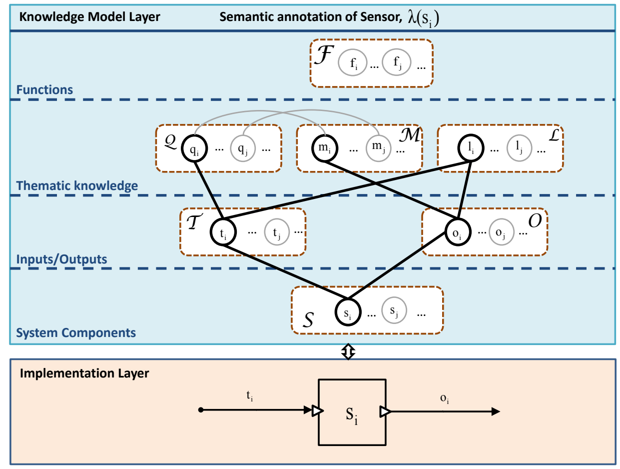


Fig. 5: The semantic annotation model of a sensor $s_i,\ i = 1,...,n_{\mathcal{S}}$ with one input $t_i,\ i = 1,...,n_{\mathcal{T}}$ and one output $o_i,\ i = 1,...,n_{\mathcal{O}}$

The components' composition algorithm first considers the types of the components and their expected role in the WDN control implementation. The position of each type of component is fixed in this work, with sensors always positioned to measure WDN outputs and then passing the measurements to a controller either directly or after processing. Then, the algorithm considers the matching of the inputs to the outputs based on their semantic properties. For instance, the location and the physical property (and/or the measurement unit) comprise important information about components' inputs/outputs. That is, the value produced by a sensor can be fed to a controller only if its location, physical property and measurement unit match to the respective properties expected by the controller. A semantic matching between an output and an input is confirmed only if *there are paths of any length from the output node to the input-node, passing through the adjacent nodes of the input at the "Thematic knowledge" layer*. The components' composition decision algorithm of the agent $\Sigma$, is given below. It is assumed that, in case of more than one appropriate controllers, these are ranked offline according to pre-defined performance criteria (the ranking is out of scope of the current stage of the work).

The execution of the algorithm will be made clearer with the illustrative cases below.

## VII. ILLUSTRATIVE CASES

Consider the configuration $I = 0$ introduced earlier and the control objective of regulating the tank head at a desired point. The composition algorithm first explores the actuators that can act on the tank head in the example WDN. Fig. 6 shows the semantic matching between the actuator $a_1$ and the plant $p_1$ = tank head regulation setup of WDN. The actuator output is annotated as deployed at location $l_2$ = the pump and producing a value in measurement unit $m_2 = meters^3$/hour. The input of the plant is annotated as representing the physical property $q_2$ = tank water inflow on location $l_1$ = tank. Since the pump position directly affects the tank head, the location $l_2$ is modeled as related to the tank location $l_1$. It can be seen that

**Algorithm 1** Components' Composition Decision Algorithm

**procedure** RUN ALGORITHM
  Find all actuators that are capable of acting on the tank water level
  If no appropriate actuators can be found, then report inability to meet the control objectives and stop, otherwise proceed to next step.
  Find all sensors that are capable of measuring the hydraulic parameters.
  **for** each available controller implementation, starting from the one with higher ranking **do**
    Check if it has the capacity to drive the actuators found in previous step.
    Exploit also available transformation functions for the control signals.
    If successful, $flag1 = 1$
    **if** $flag1 == 1$ **then**
      Find whether the sensors identified earlier measure all controller's mandatory inputs.
      Exploit also available transformation functions for the sensor's output signals.
      If successful, $flag2 = 1$
      Exit loop
    **else**
      Allow capacity to drive fewer actuators than available and continue with the next controller
    **end if**
  **end for**
  **if** $flag2 == 1$ **then**
    Match is confirmed, therefore close the loop with the matching components and continue operation of the system for the specific application.
  **end if**
**end procedure**

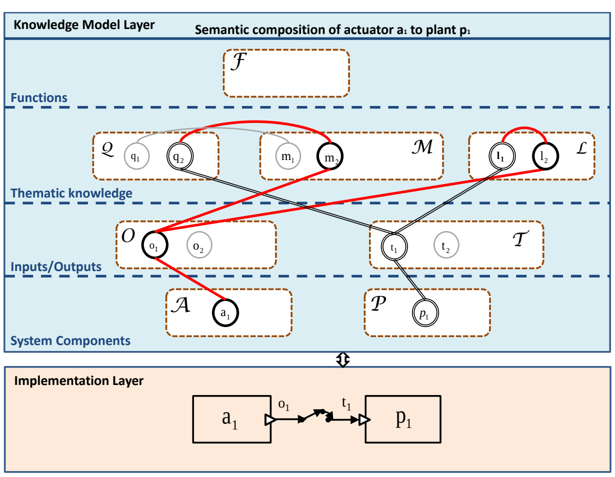


Fig. 6: The semantic matching of output $o_1$ of actuator $a_1$ to the input $t_1$ of WDN plant $p_1$

all three nodes adjacent to the input $t_1$ of $p_1$ in the "Thematic knowledge" layer, that is, nodes $l_1$, $q_2$ marked with double black lines, are reachable by paths departing from output $o_1$ of actuator $a_1$, marked with single thick red line.

The next step of the composition algorithm considers the matching between a controller and the selected actuator in previous step. The actuator is annotated as expecting an "on/off" signal (modeled as measurement unit $m_3$) in its input. Since the actuator is a physical component, the location is also inherited by its input. On the other hand, the controller is annotated as producing an "on/off" signal, while it inherits the location by the plant. This composition, as well as the composition of the plant output to the sensor input, and the one of the sensor output to the controller input are implemented in a similar way and are omitted here.

Sometime in the future, the sensor $s_1$ stopped working and had to be replaced. Since an immediate replacement was difficult to be found, the utility gave instructions to technicians to install two other sensors, $s_2$ = measuring the inflow to the tank and $s_3$ = measuring the outflow of the tank. The knowledge model has been also enriched by introducing the knowledge objects representing the two sensors ($s_2$, $s_3 \in \mathcal{S}$), the objects representing the locations ($l_3$ = pipe entering tank, $l_4$ = pipe leaving tank $\in \mathcal{L}$), an "adjacent-to" relation between locations $l_3$ and $l_4$ with $l_1$, as well as the annotations of the sensors with their relation to the physical property $q_2$ = water-flow that they and the measurement unit $m_4$ = feet that they produce. The change of the components is detected by agent $\Sigma$ (details of detection are out of the scope of this work) and it immediately checks whether a new configuration $I = 1$ can be achieved by utilizing the new components.

As far as the compositions of the available actuator to the plant and the controller to the actuator are concerned,the composition algorithm retrieves exactly the same results as for configuration $I = 0$. However, when it comes to the matching of sensor output to the controller input, this cannot be achieved using the available sensors. This is mainly because the two currently available sensors measure a different physical property. It is solved by assuming the availability of a function $f_h : \mathcal{R} \times \mathcal{R} \mapsto \mathcal{R}$, in the tools' database, which takes as input the inflow and outflow of the tank and computes the tank head, as discussed in Section II. The resulted composition of the sensors' outputs to the controller input through the function $f_h$ is illustrated in Fig. 7. It can be seen that (red-line) paths starting from the outputs of both sensors arrive to the location-annotation, as well as to the physical property annotation of the controller's input. Note that two edges starting from the measurement unit $m_2$ arrive at the function $f_h$ node to represent the two input arguments, while a single edge arrives to $m_1$ to represented the computed output. The output (a value in meters) is a measurement unit of head tank as required, therefore a semantic match is confirmed by agent $\Sigma$ for the configuration $I = 1$.

It is emphasized that the proposed architecture does not impose any limits to the complexity of the modeled thematic knowledge and reasoning mechanisms to be adopted. The examples presented here are kept simple for presentation purposes.

## VIII. Conclusions

A flexible architecture has been described, that can be adopted in the design of new generation control structures of

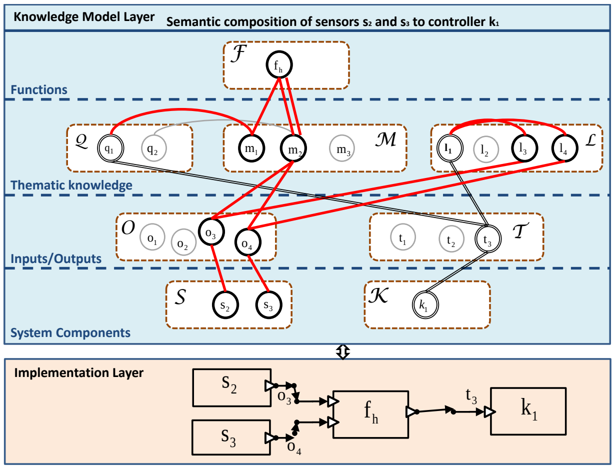


Fig. 7: The semantic matching of outputs $o_3$ and $0_4$ of the sensors $s_2$ and $s_3$ respectively, to the input $t_3$ of the controller $k_1$. The composition is achieved through the function $f_h$ that computes the tank head given the inflow and outflow of water.

WDNs, in order to take advantage of the online reconfigurability characteristics offered by the semantic interoperability and composition techniques. The current industrial practice suggests using static configurations and standard control implementations. We believe that recent advancements in smart water networks, can enormously benefit the utilities if there is a framework for the online deployment of smart components. In future steps, the flexibility of the architecture will be further improved by adopting standard knowledge modeling frameworks. In addition, the semantic composition techniques will be extended to other parts of the WDN control applications.

*Acknowledgment:* This work is partially funded by the European Research Council (ERC) under the project "Fault-Adaptive Monitoring and Control of Complex Distributed Dynamical Systems".

## References


[1] M. A. Brdys and B. Ulanicki, *Operational control of water systems: structures, algorithms, and applications*. New York, USA: Prentice Hall, 1994.

[2] D. Eliades and M. Polycarpou, "Leakage fault detection in district metered areas of water distribution systems," *Journal of Hydroinformatics*, vol. 14, no. 4, pp. 992–1005, 2012.

[3] D. G. Eliades and M. M. Polycarpou, "Water contamination impact evaluation and source-area isolation using decision trees," *ASCE Journal of Water Resources Planning and Management*, 2011, (available online).

[4] D. Eliades and M. Polycarpou, "A fault diagnosis and security framework for water systems," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 6, pp. 1254 –1265, Nov. 2010.

[5] W. M. Grayman, "A quarter of a century of water quality modeling in distribution systems," in *Proc. ASCE Water Distribution Systems Analysis*, 2006, p. 12.

[6] D. Savic, Z. Kapelan, and P. Jonkergouw, "Quo vadis water distribution model calibration?" *Urban Water Journal*, vol. 6, no. 1, pp. 3–22, Feb. 2009.

[7] E. Todini and S. Pilati, "A gradient method for the analysis of pipe networks," in *Proc. Computer Applications for Water Supply and Distribution*, Leicester, UK, 1987, p. 20.

[8] L. A. Rossman, *EPANET 2 Users manual*, EPA/600/R-00/057, National Risk Management Research Laboratory, Office of Research and Development, U.S. Environmental Protection Agency, Cincinnati, OH, Sep. 2000.

[9] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, "OWL-S: Semantic Markup for Web Services," 2008. [Online]. Available: http://www.ai.sri.com/daml/services/owl-s/1.2/overview/

[10] J. Davis and M. S. Rajasree, "RESTDoc: Describe, Discover and Compose RESTful Semantic Web Services using Annotated Documentations," *International journal of Web & Semantic Technology*, vol. 4, no. 1, pp. 37–49, Jan. 2013. [Online]. Available: http://www.airccse.org/journal/ijwest/papers/4113ijwest03.pdf

[11] A. Sheth, C. Henson, and S. Sahoo, "Semantic Sensor Web," *IEEE Internet Computing*, vol. 12, no. 4, pp. 78–83, Jul. 2008. [Online]. Available: http://knoesis.org/library/publications/SHS08-ICColumn-SSW.pdf

[12] "Sensor Model Language (SensorML)." [Online]. Available: http://www.opengeospatial.org/standards/sensorml

[13] G. Antoniou and F. V. Harmelen, "Web ontology language: Owl," in *Handbook on Ontologies in Information Systems*. Springer, 2003, pp. 67–92. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-92673-3_4

[14] G. M. Milis, C. G. Panayiotou, and M. M. Polycarpou, "Towards a Semantically Enhanced Control Architecture," in *IEEE Multi-Conference on Systems and Control*, Dubrovnik, Croatia, 2012.

[15] W3C, "Semantic Web Query Standards," 2004. [Online]. Available: http://www.w3.org/standards/semanticweb/query

[16] A. Abur and A. G. Exposito, *Power system state estimation: Theory and Implementation*, New York: Basel, 2004.

[17] M. Asprou and E. Kyriakides, "Enhancement of hybrid state estimation using pseudo flow measurements," in *Power and Energy Society General Meeting, 2011 IEEE*, Detroit, USA, 2011, pp. 1–7.

[18] T. S. Bi, X. H. Qin, and Q. X. Yang, "A novel hybrid state estimator for including synchronized phasor measurements," *Electric Power Systems Research*, vol. 78, no. 8, pp. 2452–2458, 2009.

[19] S. Chakrabarti, E. Kyriakides, G. Ledwich, and A. Ghosh, "On the inclusion of phasor measurements in a power state estimation," *IET Generation, Transmission, and Distribution*, vol. 4, no. 10, pp. 1104–1115, 2010.

[20] S. Chakrabarti, E. Kyriakides, G. Valverde, and V. Terzija, "State estimation including synchronized measurements," in *Power Tech Conference*, Bucharest, 2009, pp. 1–5.

[21] IEEE SmartGrid, "Smart Grid Experts, Information, News and Conferences." [Online]. Available: http://smartgrid.ieee.org/

[22] Z. Ming, V. A. Centeno, J. S. Thorp, and A. G. Phadke, "An alternative for including phasor measurements in state estimators," *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1930–1937, 2006.

[23] A. G. Phadke and J. S. Thorp, *Synchronized phasor measurements and their applications*. New York: Springer, 2008.

[24] A. J. Wood, B. F. Wollenberg, and G. B. Sheble, *Power Generation, Operation and Control*, 3rd ed. Wiley, 2013. [Online]. Available: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471790559.html