Fuzzy Waters: Towards a smart system middleware for wastewater networks

Jose Manuel Linares[1], François Taïani[1], David Singerton[2]
[1]Lancaster University (UK), [2]Anglian Water Services Ltd (UK)
{j.linares,f.taiani}@lancaster.ac.uk, dsingerton@anglianwater.co.uk

ABSTRACT

Water utility companies face increasing economic and environmental pressures to optimise their infrastructure, in order to save energy, mitigate extreme weather events, and prevent water pollution. One promising approach consists in using better distributed control (smart systems) in water networks. These systems are however difficult to build, and go well beyond the traditional expertise of water companies. In this paper, we argue therefore for a middleware-based approach to the construction of smart water systems. We focus more specifically on waste water systems, and review the challenges and potential approaches available to water companies. Using this analysis we then propose a high level architecture of a potential waste water middleware.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Distribution

Keywords

smart waste water systems, wireless sensor nodes, motes, SCADA, WSN, Component Based Software Engineering, fuzzy logic, distributed fuzzy logic, cyber-physical systems

1. INTRODUCTION

In most developed countries, water infrastructures and services play a critical role in protecting the public health, and preventing damages to human and natural environments. Water infrastructures are well known for delivering drinkwater, but they also play a key role in providing irrigation to agriculture, limiting water-based pollution, and mitigating extreme weather events such as floods and droughts. Managing, maintaining, and updating these infrastructures is, however, costly and complex. Water infrastructures are heavily distributed (over several 10,000 km^2 in some instances), include a large range of equipment (pipes, pumps, sewers, vanes, treatment plants, controllers), and have often been constructed over several decades, sometimes going back as far as Victorian times (1837 - 1901).

As environmental regulations progress, water consumption surges (mainly due to population growth), and energy costs rise, water companies are under increasing pressure to update and improve the management of their infrastructure. One promising plan is in using better control systems ("smart" systems) to reduce operational costs and improve services. Such systems rely on control theory (fuzzy logic, decision trees), wireless sensors and actuators networks (WSANs), and better models to achieve their aims. Unfortunately, in spite of some promising starts [10, 13], few of these systems are deployed in production today. One reason, we argue, is the lack of an appropriate integration platform (a "middleware") tailored towards the water-industry.

In this paper, we focus specifically on the case of wastewater systems. We discuss the current needs of such systems in terms of control, distribution, and integration, based on our current work with Anglian Water; one of the largest water utility companies in the United Kingdom. We then review potential solutions and current approaches to these problems, with a focus on fuzzy logic control. We finally propose some research avenues for the development of a modular and incremental middleware for distributed wastewater control.

2. BACKGROUND

Wastewater networks are complex infrastructures that combine civil engineering works (sewers, basins, reservoirs), hydraulic actuators (pumps, gates, valves), sensors (water levels and flows, toxins, gasses), and control devices (Programmable Logic Controllers). The control logic uses in wastewater networks is very simple, often relying on fixed threshold values to trigger behaviours (e.g. switching a pump on or off), but more advanced control techniques are now being considered, with fuzzy logic a promising candidate [11]. In the following, we describe in detail the structure and constituents of a wastewater network (Section 2.1). We then provide a quick introduction to fuzzy logic, and explain how it applies to the control of wastewater networks (Section 2.2).

2.1 Wastewater systems

A wastewater infrastructure is usually organised in catch-

MidSens 2012 '2012 Quebex, Canada

ments. Each sewer catchment consists of a connected network of sewer pipes that collect sewage in an area and bring it to a treatment plant or a discharge point. The number of catchments managed by a water company can be substantial and, taken together, can cover an extensive area. Anglian Water for instance collects waste water from about 6 millions customers through 1,100 waste water catchments over an area 27,500 km^2 in the East of England.

Many sewer catchments use a *combined sewer system* which collects both wastewater from households and storm water during rainfall. A combined sewer system must have enough capacity to process a large range of water inflows. This large capacity is needed to prevent toxic flooding in case of heavy waters (wet weather conditions), but also to avoid high concentrations of toxic substances if the rainwater is insufficient (dry weather conditions).

Mere gravity is usually insufficient to transport water in a sewer catchment. A catchment is therefore often equipped with a set of pumping stations that transport wastewater over an elevation, so that it can continue to flow under the effect of gravity.

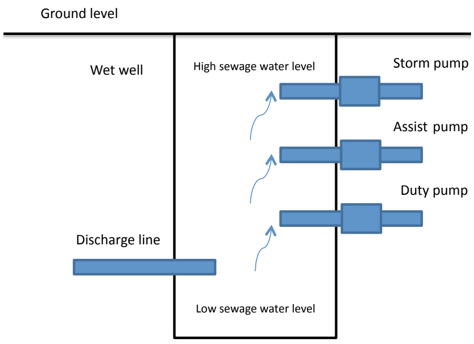

Figure 1: Schematic of a wet well

A pumping station is built around a wet well (Figure 1), an underground reservoir that acts as a buffer for incoming water from a main sewer pipe. A wet well is usually equipped with 2 to 3 pumps: a *duty pump*, an *assist pump* and, in some critical wet wells, a *storm pump* (see [11] for more details of pumps).

The pumps of a wet well can be switched on and off, and must be controlled to process the incoming water, while minimising energy consumption, and optimising the pumps' lifetime. The opportunities for energy consumption are substantial, as energy costs constitute a substantial part of the

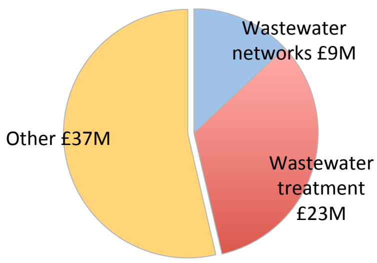

Figure 2: Anglian Water's energy costs (£60m)

operational budget of most water companies [11]. Anglian Water for instance spends about £60 million pounds in energy yearly, with £32 million was spent on wastewater operations, of which £9 million on wastewater networks (Figure 2). Similarly, unexpected pump failures can have drastic consequences, possibly leading to the flooding of pollutants. Finally, although this is rarely implemented at the moment, an advanced control of pumps holds the promise of minimising adverse events known as *Combined Sewer Overflows* (CSO for short), which are tightly regulated, and can carry heavy financial and environmental consequences.

A Combined Sewer Outage occur when the capacity of a combined sewer system is exceeded, usually as a result of heavy rainfall. Sewer networks are in this case designed to discharge some the excess water directly into the environment (river, lake or sea) without treatment. A better handling of pumps during (short) heavy rainfall could in principle mitigate CSO by better utilising the buffer capacity of wet wells.

2.2 Fuzzy logic control

In spite of the potential benefits of better control in combined sewer systems, most pumping stations are handled today by a basic logic, embedded in a Programmable Logic Controller (PLC) installed in the station. The PLC uses input from ultrasonic sensors to activate each of the pumps according to hard-coded water levels.

Any improvement on this basic approach should fit the needs of the water industry: It should have a low total cost of ownership, due to the large number of pumping stations to be equipped (Anglian Water for instance manages approximately 4,500 pumping stations which have 12,500 pump sets all together); it should be robust to unexpected events and failures; and should be easily portable to different stations with little deployment and tuning efforts.

2.2.1 *Principles of fuzzy logic*

Fuzzy logic meets most of these needs, and is one of the solutions currently researched by Anglian Water [11]. Fuzzy logic extends traditional Boolean logic with continuous truth values between 0 (false) and 1 (true), rather than just 0 and 1. In control, a fuzzy logic approach usually starts by a *fuzzyfication* step, in which the inputs of the control system are processed through a set of fuzzy membership functions [9, 7]. For instance, if one of the inputs is the water level in the wet well, Figure 3 shows the truth value of the three fuzzy predicates LOW_LEVEL, MEDIUM_LEVEL, and HIGH_LEVEL. For a water level of 1.25 meters, the truth value of LOW_LEVEL is 0.75, that of MEDIUM_LEVEL is 0.25 and that of HIGH_LEVEL is 0.

The control part of a fuzzy-base control logic typically takes the form of *fuzzy rule-based inference systems*, FRB for short. An FRB uses a set of if-then rules to encode the actions the system should take, depending on its input predicates, for instance:

`if` LOW_LEVEL `and` HEAVY_RAIN `then` ASSIST_PUMP_ON

The `if` parts (the antecedent) of all rules are evaluated in parallel, using a fuzzy semantic for Boolean operators (e.g. x `and` y is usually interpreted as $min(x, y)$). The resulting truth values are then used to compute the `then` part (consequent) of each rule, in a way that differs depending on the type of FRBs considered.

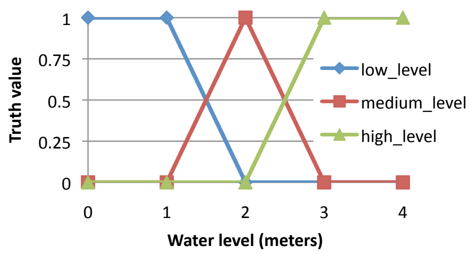

Figure 3: Fuzzy membership functions

In the simplest case (known as Mamdani FRBs), the consequent and antecedents are used to determine a fuzzy membership function for the whole FRB through an *implication* and an *aggregation* step. This function computes how the truth value of the whole system varies with the output variables (in our case the switching states of the pumps) for the observed input (the water level).

The final stage consists then in transforming this system-wide membership functions into actual control values (which pumps to switch on/off), known as the *defuzzyfication* step. This last step usually seeks to maximise the average truth value of the system, while taking into account the many-value logic captured by the FRB.

Although all FRB follow the above steps (*fuzzyfication*, *implication*, *aggregation*, *defuzzyfication*) variations point are numerous. First, Mamdani inference systems are only one type of FRBs, with Tagani-Sugeno FRBs another important alternative. Tagani-Sugeno FRBs use linear combinations of antecedent truth values in the *implication* and *aggregation* steps to directly provide an output value for control. Within the Mamdani family itself, many variants exists depending on the semantic of the fuzzy operators (*max* being one option only for `and` for instance), and the detail of the aggregation and defuzzyfication steps.

Besides the above design choices, the quality of control delivered by an FRB strongly depends on the design of its input membership functions. Here again several approaches exist: A basic approach consists in relying on expert knowledge. More advanced strategies are however often taken that use optimisation to search for an optimal set of membership functions maximising particular quantities (energy consumption, pump lifetime, risk of CSOs). This optimisation usually occurs off-line using historical data, but recent works have been proposed to optimise membership functions dynamically based on the system's feedback [12].

2.2.2 *Fuzzy logic at Anglian Water*

Anglian Water have started experimenting with fuzzy-logic to optimise the energy consumption of one pumping station in dry weather condition [11]. The approach uses a Mamdani FRB to switch the pumps of the wet well on and off depending on the wet well's water level, the flow of water coming in, and the current electricity tariff (night or day). The input membership functions are configured off-line on historical data using a genetic algorithms to optimise the energy consumption, while minimising pump switching (a factor of premature wear).

This early system was shown to deliver energy saving of between 2 and 2.5%, a promising start which is now being refined by Anglian Water (at present, it is over 6%).

2.3 The need for middleware

Although Anglian Water's experiment with fuzzy logic control is promising, most of the benefits of more advanced control require a distributed design [13, 10]. In particular, a more fine-tuned approach to pump control is only possible with more coordination and more distributed data. Minimising CSOs for instance will require the coordination of several pumping stations in a catchment, and the use of more sensed data obtained from wireless sensors in the pipe themselves.

Designing an implementing such distributed solutions go beyond the expertise of most water companies, and requires to blend in both distribution and control concerns. Because sewer infrastructures are long-lived systems (many decades), any solution also needs to allow incremental changes and updates, both in terms of hardware and software, while providing enough flexibility to be adapted to the specifics of each individual deployments.

All these constraints suggest the need for a modular platform providing both distribution and control to water utilities while insulating them from the intricacies of WSN/PLC integration, multi-station coordination, sensor value sampling, and fail-over techniques. In the following section, we review the state of the art in terms of middleware for water networks before analysing the research challenges arising from these early attempts (Section 4), and sketching a possible architecture for such a platform (Section 5).

3. EXISTING PLATFORMS

We now review existing systems that address some of the problems discussed on the previous section. We discuss SCADA, SWATS, PIPENET and other new research ideas that are emerging and what each system's strongpoints and weakpoints are in our context for a controllable distributed system.

3.1 SCADA systems

A type of communication system that is being used within the waste water and oil industries is SCADA (Supervisory control and Data Acquisition) [15][13]. SCADA systems are typically centralised systems where sensors (sensing data from actuators) sends all the data to a central head quarters where a human operator monitors the data which presents itself as alarms. Communications within the SCADA network involves using radio, POTS (Plain Old Telephone Service) leased lines, cellular (GSM) or sattelite communications. Unfortunately, SCADA has several shortfalls [15] which is that the equipment is expensive (proprietary technology), unscalable, suffers from high latency in sending data, inflexible in software use, and unable to be fully distributed as it is a centralised architecture. Yoon et al [15] have proposed and developed a system which is specifically built to monitor high pressure water, *waste water*, and oil flows though it does offer some advantages over SCADA and offers insights in proposing a system suitable for the wastewater industry and to control flow of wastewater from the wet wells.

3.2 PIPENET

PIPENET [14] is a WSN network that was tested within a 22 month trial period in Boston, US, that monitor pipelines to detect, localise and quantify bursts and leaks in water

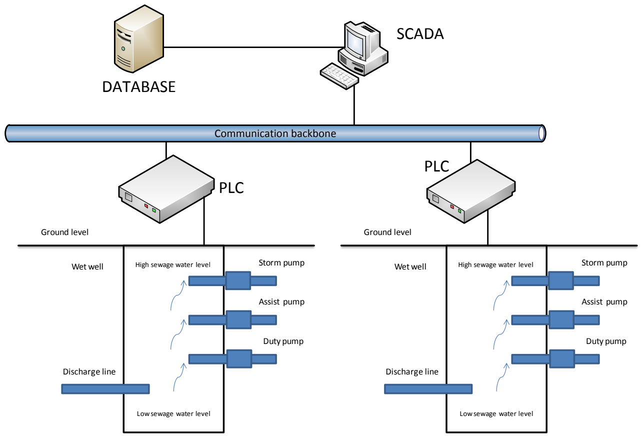

Figure 4: Simple SCADA diagram in the water utility industry

transmission pipes and the monitoring of water levels in sewer collectors and CSO's. The system also reports anomalies in pipelines such as blockages and malfunctioning control valves and also monitors levels in sewer collectors. The deployment that consists of Intel Mote platform [1] is a centralised network where data is sent from the nodes to the backend servers and require substantial computing power. Although PIPENET has support for high data rate time synchronized data collection, it can not support high levels of local data processing as the motes are specifically designed for this application. The motes [14] consist of an ARM7 core, 64kB RAM, 512kB Flash memory and a bluetooth radio that are deployed on the first tier which is the sewage pipe system, the 2nd tier is the gateway which is essentially the head of a cluster of motes and sends data from the motes to the backend via GPRS or EDGE. The third tier is the backend where the data is recieved from the gateway and data is stored for offline analysis. Although PIPENET was an approach to replace SCADA, PIPENET lacks the facility to provide in-network processing algorithms and we shall look at into an alternative [15] which we will discuss in (Section 3.3).

3.3 SWATS

We have seen that SCADA and PIPENET are two separate systems but has its weaknesses. Yoon et al [15] has been involved in the development of an emerging application called SWATS (Steamflood and Water flood Tracking System). Previously, the SCADA system had the role of detecting anomalies and identifying the locations of problems (e.g. leakage or blockage in pipes), and now, a fully distributed approach (SWATS) is realised as it addresses the shortfalls of the SCADA and PIPENET systems: it is low cost, low latency, fine granular coverage, highly accurate and reliable. In SWATS, WSN nodes are cheap to acquire and are able to collaborate with other nodes to improve correctness: a method known as multi node - multi modal that detects leaks or blockages within the pipe network by comparing data from other nodes, and exploits temporal and spatial correlations from past sensor readings by means of a Decision Tree Algorithm (DTA). We will briefly explain the details in (Section 3.4.1).

3.4 Past research and development

The challenges that are current in distributed systems range from communications, radio signal failures to hardware resource limitations on the actual sensor motes. There have been attempts by researchers to develop heterogeneous WSN systems and middleware software that is able to perform in a prescribed environment. Such environments can be oil fields where sensors takes data samples of the health of an oil pipe to a water catchment where ultrasonic sensors are used to monitor water levels within a wet well.

3.4.1 Fault tolerance in sensor data

Marin-Perianu et al [8] highlights that a single sensor will not guarantee reliable data received from sensor inputs. SWATS addresses this problem using a Decision Tree Algorithm to make comparisons over which alarm to trigger and hence avoid false alarms. To determine the causes and the health of a pipeline, the DTA uses historical data stored in the central database and from the reported events from the other neighbouring nodes and information to the proximity to the equipment in question. D-FLER [8] is a distributed, general purpose reasoning engine for WSN. This approach uses fuzzy logic to make comparisons with observations with neighbourhood nodes for detecting events and produces a reliable and accurate result. D-FLER was confined within a fire prevention network simulation with promising results as false fire alarms was less than 2%.

3.4.2 Other Middleware

In terms of middleware, Iqbal et al [6] proposes a middleware that is designed to monitor large scale water distribution systems. The system itself is event driven (such as leaks, pipe burst, contamination and security), specialised and reconfigurable. The author Rocha, et al [4] proposed a semantic middleware for an Autonomic Wireless Sensor Network (AWSN) with the vision of being applied to Structural Health Monitoring applications (SHM). Although its main focus is in SHM areas, it can be used for other areas such as ambient intelligence, habitat monitoring and fire detection. It uses the concept of fuzzy logic to create fuzzy rules in order to compose a knowledge base for the domain.

We have seen three different types of systems and some other proposals. Although researchers are stressing the need for modular and flexible middleware for existing infrastructures, such as in our case, the waste water industry, we have still not observed research that focuses on these properties with the addition of control.

4. CHALLENGES IN WASTEWATER DISTRIBUTED SYSTEMS

There are several key challenges in distributed systems that range from the lowest communication physical level to the application level where a software system requires to be heterogeneous with hardware motes.

4.1 Real time control

Schutze, et al [13] relates that with today's available technology and methodologies, it is possible to allow real time control (RTC) of urban wastewater systems. RTC is a desirable property in wastewater systems as it allows a better control when problems happen. For example, if a wet well is flooded, the central control would like to know of this situation on the moment it (ideally < 1 minute) otherwise an area would be flooded without the operator's knowledge. The author also emphasises that there is a need for clear terminologies in RTC so that scientists and experts from other domains can collaborate when developing an RTC system.

4.2 Decentralisation, sensor accuracy and deployment

At present, Anglian Water have a centralized system which is not fault tolerant as the central computer manages all communications and there is a single point of failure. Having a decentralized solution eliminates a system wide failure if one single component would fail.

A key challenge in Anglian Water is to reduce the rate of false alarms that go through the main system as human operators can not be scaled to monitor thousands of alarms being logged. In section 3.3, Yoon, et al [15] worked on DTA where nodes collaborate with each other to reduce false alarms. Fault tolerance is a topic discussed in section 3.4.1 as it was stated that no single sensor is completely reliable.

Deployment of sensors is usually done in difficult or hazardous areas such as sewer systems and wet well chambers. Some reasons to physically access a sensor include replacing batteries, or replacing the unit, thus it is desirable that the nodes deployed requires as little human intervention as possible as in our case study, the sensors will be deployed in the afore mentioned areas.

4.3 Distributing intelligence across sensor motes

Ostojin's [11] results are promising for a small scale catchment area. However there is no mention of how such a system could scale up to a catchment that has more than five pumps in future work. There is concern that because Mamdani rules have to be created manually, it can add extra burden on the domain expert. Furthermore, the rules given are computationally expensive which means that a small WSN mote will not have the CPU capability to process the rules, hence there is a possibility that a hybrid approach in distributing intelligence is required to process hard calculations.

5. TOWARDS A MIDDLEWARE FOR WASTE WATER NETWORKS

We now explore how a middleware framework for smart waste water networks might look like, on the basis of our previous discussion. We propose a component-based design, a software technology that has been shown to work well for embedded distributed systems [3, 5]. A key goal of our design is to deliver a *white box* middleware that can be changed and adapted easily by potential users (e.g. water utility companies) to fit their own needs, while providing high-level abstractions suited to waste water networks.

5.1 Physical architecture

The physical organisation of the system is shown in Figure 6. The system is organised in three levels: headquarters (central box), catchment areas (left and right), and pumping stations (smaller boxes). This choice reflects the need to integrate legacy alarm and CRM (Customer Relationship Management) systems in use in many water companies. These systems are typically hosted centrally, at the company's headquarters, and need to be fed with information from the field. Water companies further need to log the activity of their network for audits and historical analysis, a function usually performed by a large central database.

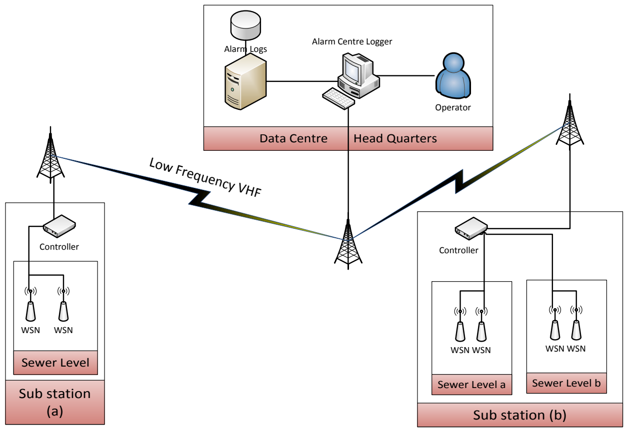

Figure 5: Envisaged high level distributed smart system

In our envisaged design, each pumping station is equipped with a controller in charge of managing the station's pumps. Stations communicate with each other and with sensor nodes deployed in the sewer pipes using low-band VHF, with one station elected at the coordinating lead for each catchment. The sewer sensors provide readings on the current water levels and inflows in the pipes which are sent to the station's controller. Status updates and alarms from a catchment (e.g a node running low on power or lost connections with a node) are sent back to the headquarters for processing.

5.2 Middleware architecture

Our envisaged middleware architecture is hierarchical and mirrors the physical organisation we have just described. It is organised in three main layers, shown in figure 6. The first layer provides connection with the centralised back-end system, where alarms and system status data are logged. Logged alarms are processed by human operators who coordinate field interventions. The second layer is decentralised coordinating the pumps and actuators within a catchment area, and reports alarms and equipment status data to the first layer thus allowing operators to remote control pumps and sensors from the headquarters (if required). The separation between the two layers is needed to accommodate the typically poor quality of communication links between a company's headquarters and its catchments. In the case of Anglian Water, this communication is currently provided by GPRS links, with one third of the links usually failing temporarily in wet weather conditions.

Finally, the third layer works at the level of a pumping station. It manages the readings provided by sensors deployed in the pipes, and provides them to the station's controller. These readings are fed into the controller's fuzzy engine to control the pumps, and trigger alarms.

5.3 Controller component layer

All layers are component-based. For place reasons we only describe the organisation of the controller layer shown on Figure 7. This figure shows the component structure of the

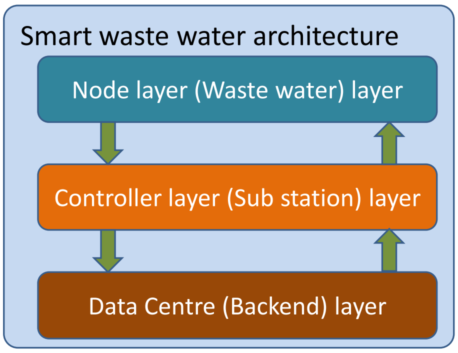

Figure 6: Smart Waste Water System Architecture

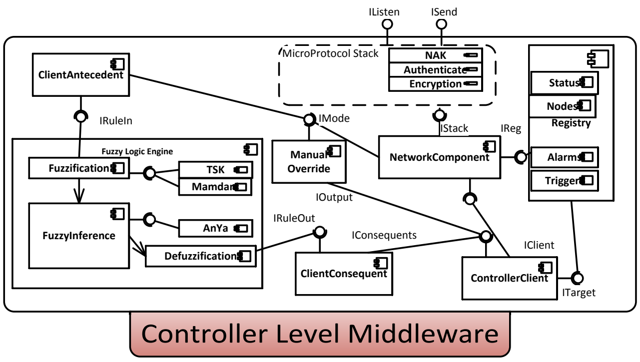

Figure 7: Component diagram at the controller layer

controller of each pumping station. Boxes represent components; edges ending with a circles are provided interfaces; edged ending with a semi-circle are required interfaces.

The server component listens for and serves requests from distant nodes (other controllers, headquarters, sensor nodes). The input types received can be registering a node, receiving antecedents (or inputs) for one of the fuzzy rule base system (Mamdani, TSK) or requesting a status update as the main head centre polls the controller. Following a client - server pattern to serve different components, the micro protocol stack provides desirable non-functional properties such as security and fault-tolerance [2]. The controller client forms the messages by combining inputs from the consequents which are the output of one of the configured fuzzy rule base systems and the properties of the message from the microprotocol stack. The registry component has the responsibility of checking whether the nodes are reachable. If a node is not reachable, an alarm is generated and it is sent via the trigger sub component to the main head centre so that the alarm is logged into the system.

The fuzzy-based engine itself is component-based, to maximise reuse, allow change and evolution, and provide good portability between different hardware platforms.

6. CONCLUSION

In this paper, we have discussed some the challenges involved in developing better distributed control systems for "smart" waste water networks. Our findings argue strongly for the development of a middleware-based approach to help the water industry design, deploy and maintain these systems. To support this claim we have presented an early component-based design of such a middleware that we now plan to prototype on a real field deployment in the near future.

7. ACKNOWLEDGMENTS

This work is being supported by Anglian Water Services Ltd and the EPSRC within the iCASE project entitled: *Building Tomorrow's Smart Waste Water Networks*.

8. REFERENCES

[1] ARM7 processor family http://arm.com/products/processors/classic/arm7/.

[2] K. P. Birman, R. Constable, M. Hayden, J. Hickey, C. Kreitz, R. Van Renesse, O. Rodeh, and W. Vogels. The horus and ensemble projects: Accomplishments and limitations. Technical report, Ithaca, NY, USA, 1999.

[3] G. Coulson, G. Blair, P. Grace, F. Taiani, A. Joolia, K. Lee, J. Ueyama, and T. Sivaharan. A generic component model for building systems software. *ACM TOCS*, 26(1):1–42, Feb. 2008.

[4] A. R. da Rocha, F. C. Delicato, J. N. de Souza, D. G. Gomes, and L. Pirmez. A semantic middleware for autonomic wireless sensor networks. In *Proceedings of the 2009 Workshop on Middleware for Ubiquitous and Pervasive Systems*, WMUPS '09, pages 19–25, New York, NY, USA, 2009. ACM.

[5] D. Hughes, K. Thoelen, W. Horré, N. Matthys, J. D. Cid, S. Michiels, C. Huygens, and W. Joosen. Looci: a loosely-coupled component infrastructure for networked embedded systems. In *7th Int. Conf. on Advances in Mobile Comp. and Multimedia*, MoMM '09, pages 195–203, 2009.

[6] M. Iqbal and H. B. Lim. A cyber-physical middleware framework for continuous monitoring of water distribution systems. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 401–402, New York, NY, USA, 2009. ACM.

[7] R. L. J Yen. *Fuzzy Logic: Intelligence, Control, And Information.* Pearson Education, 1999.

[8] M. Marin-Perianu. D-FLER, A Distributed Fuzzy Logic Engine for Rule-Based Wireless Sensor Networks. *Ubiquitous Computing Systems*, (1):86–101, 2007.

[9] J. Mendel. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 83(3):345 –377, mar 1995.

[10] L. Montestruque and M. Lemmon. Csonet: a metropolitan scale wireless sensor-actuator network. In *Int Workshop on Mobile Device and Urban Sensing (MODUS)*, 2008.

[11] S. Ostojin, S. Mounce, and J. Boxall. An artificial intelligence approach for optimizing pumping in sewer systems. *Journal of hydroinformatics*, 13(3):295–306, 2011.

[12] P. Sadeghi-Tehran, A. B. Cara, P. Angelov, H. Pomares, I. Rojas, and A. Prieto. Self-evolving parameter-free rule-based controller. In *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE Int Conference on*, pages 1 –8, june 2012.

[13] M. Schütze, A. Campisano, H. Colas, W. Schilling, and P. a. Vanrolleghem. Real time control of urban wastewater systems, where do we stand today? *Journal of Hydrology*, 299(3-4):335–348, Dec. 2004.

[14] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline. Pipenet: a wireless sensor network for pipeline monitoring. In *6th Int. Conf. on Information Proc. in Sensor Networks*, IPSN'07, pages 264–273, 2007.

[15] S. Yoon, W. Ye, J. Heidemann, B. Littlefield, and C. Shahabi. Swats: Wireless sensor networks for steamflood and waterflood pipeline monitoring. *Network, IEEE*, 25(1):50 –56, january-february 2011.