Bringing simulation to engineers in the field: a Web 2.0 approach

BY ROBERT HAINES 1,*, KASHIF KHAN 2 AND JOHN BROOKE 2

1 Research Computing Services, Devonshire House, and 2 School of Computer Science, Kilburn Building, University of Manchester, Manchester M13 9PL, UK

Field engineers working on water distribution systems have to implement day-to-day operational decisions. Since pipe networks are highly interconnected, the effects of such decisions are correlated with hydraulic and water quality conditions elsewhere in the network. This makes the provision of predictive decision support tools (DSTs) for field engineers critical to optimizing the engineering work on the network. We describe how we created DSTs to run on lightweight mobile devices by using the Web 2.0 technique known as Software as a Service. We designed our system following the architectural style of representational state transfer. The system not only displays static geographical information system data for pipe networks, but also dynamic information and prediction of network state, by invoking and displaying the results of simulations running on more powerful remote resources.

Keywords: water distribution systems; Web 2.0; representational state transfer; simulation; decision support tools; geographical information system

1. Introduction

Water distribution systems (WDSs) are among the largest infrastructures spread all over the country.1 Owing to their scale and the age of the pipe network, existing WDSs encounter day-to-day operational problems. These include leakages in pipelines, malfunctioning of control valves, discoloration and accidental or intentional contamination of water. Since a typical pipe network in a WDS is highly interconnected, a problem at a single point in the system may rapidly affect a large portion of the network. For example, a leakage in a pipe may cause the contamination to enter into the network and spread rapidly via different paths in the network. To minimize safety impacts and to ensure service delivery, water utilities are therefore challenged to provide quick and efficient solutions to the problems encountered in WDSs.

Field engineers of water utilities are responsible for localizing and rectifying the WDS problems in order to keep the system operational. To provide site-specific local knowledge, field engineers are now typically equipped with a

* Author for correspondence (rhaines@manchester.ac.uk).

1 Over 700 000 km of pipes, 160 times greater than the UK’s motorway network.

One contribution of 16 to a Theme Issue ‘Crossing boundaries: computational science, e-Science and global e-Infrastructure II. Selected papers from the UK e-Science All Hands Meeting 2008’.

This journal is © 2009 The Royal Society
Toughbook (rugged laptop) running a customized geographical information system (GIS) application developed using commercial off-the-shelf GIS tools such as ArcGIS (Esri 2008) or MapInfo (2008). Current tools provide static GIS datasets of the operating sites that help engineers to analyse and understand the geographical structure and attributes of the pipe network in order to prepare the rectification plan. However, implementing the rectification plan involves altering the operational network (e.g. closing a valve) in order to isolate, for example, leaks/bursts. Each alteration has a considerable impact on the hydraulic and quality aspects of the water network, e.g. pressure loss or discoloration (Mays 2000); therefore, it is necessary to evaluate the possible consequences of each alteration before performing them on the physical network. This highlights the need to have dynamic information about the functionality of the network, such as hydraulic conditions (e.g. pressures), along with the existing static GIS functionality available to the field engineers.

Hitherto the simulation models have been run in central control rooms by specialist modellers, and the user interface to the simulation has been tailored to their expert needs. Field engineers require answers to more constrained questions and do not have specialist knowledge of the model. The challenge is therefore to present the results of specific queries obtained by running a simulation to engineers in the field in a relevant form, displayable on lightweight devices connected to the servers running the simulation via wireless networks. The problem of using lightweight devices to present the results of large simulations has received attention from the e-Science community (e.g. Holmes & Kawalsky 2006; Parkin & Brooke 2007). The former work used the three tier architecture of UNICORE and the latter used Microsoft .NET services. Both solutions relied on specific server-side services, now based on the Web Services Resource Framework standard.

It has recently been proposed that Web 2.0-style (O’Reilly 2005) distributed computing can be used as an alternative to or to supplement existing grid computing approaches for e-Science (Pierce et al. 2006, 2007; Fox et al. 2007). Web 2.0 is a somewhat overloaded term that can refer either to user-generated and/or managed online content or the provision of applications via the Web (O’Reilly 2005). The latter is commonly referred to as Software as a Service (SaaS). These authors (and others) observe that SaaS allows scientists to develop applications in an agile and iterative fashion using technology made familiar by the global scope of companies such as Google and Amazon. Our own method uses a document-based approach whereby all communication between client and simulation is via messages encoded in extensible markup language (XML) and transmitted by the hypertext transfer protocol (HTTP).

The work presented in this paper uses these methods to build a lightweight and interactive software solution to provide decision support tools (DSTs) for field engineers operating on WDSs. This is achieved by invoking and displaying the results from an existing hydrodynamic modelling library, EPANET2 (Rossman 1994), in a Web-browser-based client/server application. The remainder of this paper describes in detail our approach and is structured as follows. First §2 discusses the architecture of the system. Section 3 then describes the features of the system and §4 discusses the system evaluation and user feedback. Finally, §5 discusses the status of our prototype and how it might be used more widely in field engineering.
2. System architecture

The architecture for our prototype system is influenced by two considerations: the requirements of providing water network simulations to engineers in the field, and the construction of a generic framework that can be used to help with decision-making in similar domains, e.g. electrical power grids and emergency response scenarios in case of fire or flooding. The following subsections describe the software model, tools and technologies that are used to develop the system architecture.

(a) Lightweight client model

We selected the Web-browser-based client/server model as the high-level architecture for our prototype system. In order to provide a lightweight client, the server is responsible for performing intensive computations, such as hydraulic simulations, and managing data repositories. The client is only responsible for sending requests to the server and the retrieval and display of results.

The Web-browser-based client/server model provides several benefits. Most importantly, it separates the client and server, allowing the server-side code, simulation models and data resources to be modified and upgraded without disturbing the client implementation. In addition, moving the simulation functionality away from the client provides the opportunity to calibrate the simulation model with the decisions taken by many engineers, enabling simulations to perform accurate predictions with an up-to-date view of the data.

(b) Web 2.0

The implicit request–response-based interactions in our Web-based architecture raises the question of how to provide an interactive, consistent and responsive user interface. Another important issue is the provision of a map-based graphical user interface. These features are necessary in order to allow field engineers to interactively prepare ‘what-if’ simulations, and to visualize and interpret the generated results.

Much of what we wanted to achieve could be done using Web 2.0 techniques. Our prototype system falls into the category of SaaS at present, but it is our intention that engineers will be able to upload details of operational changes that they have made to the network to update the central model, thereby completing the interactive loop. The Web 2.0 elements of our prototype implementation are now described.

(i) Asynchronous JavaScript and XML

An interactive, consistent and responsive user interface is required by the field engineers, especially while analysing the structure of the water network or interpreting the time period forecast of the network simulations. To provide such an experience to the user within the constraints presented by a Web browser, on potentially slow and high-latency network links, we needed to perform much of the data transferral asynchronously so that the interface remained responsive. To achieve this, we used a set of commonly available functions provided by the JavaScript interpreters of the major Web browsers that are collectively referred to as asynchronous JavaScript and XML (AJAX; Perry 2006).
The AJAX functionality provides the client with the ability to perform both communication with the server and XML (W3C 2006) parsing in the background, while the user continues to interact with the interface as normal. This means that dynamic content, such as simulation results, can be requested, received and interpreted without interrupting the behaviour of the current page. Appropriate views of the data are generated dynamically from the pre-downloaded extensible hypertext markup language and cascading style sheets code by processing the XML responses from the server.

(ii) Google maps

The map-based interface is necessary to provide the representation of the physical water network in the client application. The geographical placement of the pipe network on top of the street map makes the structure of the pipe network easily understandable and helps field engineers to locate the problematic areas and to prepare the rectification plans. In addition, owing to the geographical placement of network components, engineers can easily perform the proposed alterations to the status of the network components (e.g., closing a valve) for different what-if experiments. Furthermore, displaying the results as colour-coded overlays on top of the map helps the engineers to easily interpret the predicted network response.

We selected the Google Maps Application Programming Interface (API; Google Maps 2008) to integrate maps and mapping functionalities, such as zooming and panning, into our client application. This freely available JavaScript-based API provides the functionality required to overlay colour-coded polylines and markers, used to represent pipes and junctions, respectively, on its street map display. All the overlay data are positioned using standard latitude and longitude coordinates so that when the user zooms or pans the view, the Google Maps code automatically adjusts the overlays to ensure their accuracy.

(c) Server

As discussed in §2a, the server is responsible for performing intensive computations, such as hydraulic simulations, and for managing data repositories, e.g. water network description files. In order to perform hydraulic and water quality simulations, we selected the EPANET2 (Rossman 1994, 1999) simulation engine, which is a freely available water distribution modelling package from the Environmental Protection Agency (EPA) in the US. Its well-defined C API allows us to incorporate the simulation engine into our prototype system to perform extended period simulations of hydraulic and water quality behaviour within the selected pipe network.

The EPANET2 input file format allows us to specify the pipe network topology, link/node physical characteristics and the initial conditions of the network. This information is then used by the simulation engine to simulate the space–time variations of flows, pressures and water quality concentrations using well-established principles (Walski 1984). However, several modifications have been made to the EPANET2 API so that it can be used with our server-side implementation. For example, a Perl wrapper for the EPANET2 simulation engine has been generated to execute simulations from the server-side Perl scripts.

Phil. Trans. R. Soc. A (2009)
The server-side code all runs within the Apache HTTP Server (Apache 1995). This provides a reliable and secure container that is thoroughly tested against high network loads.

\[(d)\] Representational state transfer and hypertext transfer protocol

Representational state transfer (REST) is an architectural style that treats resources as nouns that can be manipulated by a selection of verbs (Fielding 2000). The Web is an example of a RESTful system that represents its resources as uniform request indicators (URIs) and operates on those resources using the HTTP methods head, get, put, post and delete (W3C 1999). When designing RESTful Web Services, care must be taken to ensure that the resources really are nouns, rather than verbs.

Furthermore, a RESTful Web Service must be hypertext driven (Fielding 2008). Hyperlinks should be included in resources to provide navigation paths through the system. In our system, for example, the client downloads a list of discrete metering areas (DMAs) that the server has data for. This list is presented to the user as a drop-down menu from which they can select the DMA of interest. The client then requests the data for the selected DMA. Figure 1 shows a typical reply from the server when the client requests details of a DMA with the hyperlinks given in XML Linking Language (XLink) format (W3C 2001).

In this way, the client can continue drilling down into the DMA data by following hyperlinks in each resource.

To upload data to a particular URI on a server, one would use the HTTP put method, but our system operates on the basis of uploading data to be processed by the server. To do this in HTTP and REST, the post method should be used (W3C 1999). In our system, the engineer is able to run a hydrodynamic analysis, either on the unchanged network or after they have altered various network parameters. If running on the unchanged network, a get request is made, e.g. http://example.com/dma/j796/hydro, and if running on a changed network, a post request is made to the same URI with the changes, XML encoded, in the message body.

While implementing this system, we have found the RESTful style to be more flexible than SOAP/Web Service Description Language and development time was reduced. The simple HTTP interactions between components were less complex to design and debug than we expected, given our experience of building SOAP-based Web Services. A further advantage we have found is that, compared with SOAP, the RESTful style presents a lower barrier to entry for people with no previous experience of Web Services.
3. System features

As previously discussed, the main aim of our proposed system is to help engineers in decision-making during work in the field. Our prototype implementation fulfils this aim by providing a lightweight solution for bringing both static and dynamic network information to the field engineers. The following sections briefly highlight key features of the developed system.

(a) The network analysis panel

The network analysis panel, as shown in figure 2, is used to provide geographical pipe network information to the field engineer. Colour-coded polylines and marker overlays are used to draw different network components (e.g. pipes, junctions or reservoirs), which help the engineer to identify and locate the appropriate network components during analysis. Different map views (e.g. satellite or street) and zoom levels can also be selected via controls provided by the Google Maps API. In addition, hydraulic and water quality attributes of the network components can be analysed by either selecting the components on the map or their names in the list boxes available in the widgets section of the panel. Furthermore, network components of a particular type can be removed from the map by selecting the toggle hyperlink preceding the component type name in the widgets section. This allows the engineer to customize the map interface according to his needs.

In addition to providing site-specific geographical knowledge, the network analysis panel also allows the field engineer to set up and initiate simulations for different what-if experiments to evaluate system response against alterations. The status of network components can be altered by selecting the components on the map or their names in the list boxes. Then, either hydraulic or water quality simulations can be executed by pressing the corresponding buttons available in the panel.

(b) The simulation analysis panel

The simulation analysis panel uses the same map view as that in figure 2, but is used to present the time period forecast of the selected simulation, either hydrodynamic (e.g. pressure and flow) or water quality. As with the network analysis panel, the simulation results are presented in the form of colour-coded overlays. Appropriate legends are provided to interpret the colour coding and consequently the hydraulic or water quality results at each of the network components. Predicted network behaviour at different time periods can be analysed by selecting the time interval from the ‘time step’ control.

The hydraulic simulation functionality in our prototype system allows engineers to easily identify those decisions that may cause infrastructure damage by generating high pressures or those that may result in service unavailability due to the significant loss of pressure. By doing several what-if experiments, engineers can find the best possible decisions that imply minimal disturbances in the operational network.

Our system also provides access to the three types of water quality analyses supported by the EPANET2 simulation engine: age; source trace; and chemical concentration analysis. In the case of ‘bad taste’ complaints from customers, age analysis can be used to predict the age of water in the network for a stipulated time period. The results can then be used to identify areas in which the age of water goes...
beyond the recommended safety limits. Similarly, in case of possible alterations in
the network (e.g. closing a pipe), age analysis can also be used to identify the impact
of such alterations on the age of water in the operational network.

(c) Client implementation for handheld devices

In order to provide pervasive access to the developed system, we have imple-
mented the client application for personal digital assistant (PDA) and smartphone
devices. Owing to our RESTful client/server design, only the interface layout needed
to be adjusted to fit it to the limited screen of such devices, as shown in figure 3.
The system can thus be accessed from a handheld device via any recent mobile
browser capable of supporting Web 2.0 technologies, such as Opera MINI v. 3.0.

4. System evaluation

(a) Preliminary evaluation

Preliminary evaluation of the prototype system is performed in the laboratory by
using water network data provided by Yorkshire Water Ltd and United Utilities
Ltd. The data contain the network topology, link/node physical characteristics
and initial conditions for two physical pipe networks, one in Sheffield and another
in Cheshire. This provided us with hydraulically balanced networks, which is a
basic requirement of the EPANET2 simulation engine and therefore critical for
validating the results of extended period hydraulic and water quality
simulations. These data are used in the testing of the system by running
hydraulic and water quality simulations, both from the desktop and from the

Figure 2. The network analysis panel.
PDA client. The same simulations were then carried out in the EPANET2 standalone graphical user interface by using the same configuration and network description files, and the results were then verified with the results of our system. In addition to this server-side testing, the real data also provided us with the provision for testing client-side functionalities. This includes displaying the correct geographical area, network topology and the link/node characteristics in both the desktop and the PDA interfaces.

(b) User feedback

After verifying the functionalities of the overall system, we gave a demonstration to Yorkshire Water Ltd and United Utilities Ltd in order to acquire expert user feedback. Several interface-related issues have been identified by the participants in order to make the system more helpful for the field engineers. For example, it has been suggested that we include flow direction indicators on the network links (e.g. pipes) while analysing the time period forecast of the network simulations. The most important suggestion, on which we are currently engaged, is to calibrate the centralized simulation model with real-time readings from sensors deployed in the field. This would allow the simulation model to be kept up to date with the state of the network resulting in real-time accurate predictions from the simulations.

(c) System limitations

There exist several limitations in our prototype system, most of which have been noted above. In addition, the engineers working in different parts of a physical WDS cannot presently coordinate their activities before altering the network. This will be necessary as two decisions can cause different actions when performed in a different order due to the highly interconnected network.
The absence of coordination may result in control decisions that appear safe in local context, but may cause conflicts globally with the decisions of engineers working in other parts of the water network. This can be solved by maintaining and incorporating possible future conditions of network components (safe control decisions planned by the field engineers) into the concurrent what-if simulations.

As with all such distributed systems, there is a very real danger that connectivity may be lost, either between the client and server or between the two components of the server. The use of the HTTP protocol means that our system can be run over any network, via a standard virtual private network for security if required, should the one in use fail. Distributing and, more importantly, replicating server components across multiple grid infrastructures will minimize the risk of disconnection while also providing scalability.

5. Lessons and future work

This paper has described a prototype decision support system that we developed for water utilities field engineers. The key difference in our work and the existing applications is that our work not only provides static GIS functionality, but also offers dynamic information about hydraulic and water quality conditions in the network through network model simulations. The current implementation is a prototype for the purposes of demonstrating the concept. The major lesson we have learnt has been that such DSTs need to be integrated into system-wide engineering practice. A key example of this is that the EPANET simulation model is not currently adapted to take up-to-date information on the state of the system, derived from sensor networks and field operative reports. We are currently working on an extension of the project to develop an extended architecture that could also enable the simulation model to be dynamic and responsive to the current state of the system.

This highlights an interesting distinction between the application of e-Science methods in scientific investigation and in engineering practice, what one might call ‘e-Engineering’. In the latter, considerations of real-time decision-making come to the forefront. The provision of dynamic field support tools raises questions about the development of current network models as they are integrated into field engineering practice. It also raises questions about the operating methodology of a large company and workforce that have social and cultural dimensions, as well as technical ones.

In future research, we plan to improve the architecture of the server running the simulation model in order to support the real-time calibration of the model via a sensor network and to execute computationally intensive simulations. For this purpose, we are planning to enable our simulation services to further access and use grid infrastructures. This will allow our system to execute computationally intensive and parallel simulations that will result in faster responses to simultaneous requests from multiple clients. This raises issues of concurrency in the total distributed system in which the field engineers and their clients become an integral part. For example, supposing that two engineers are about to take decisions about network engineering, each of which would impact on the other. We then have classic issues of ordering and determinancy, which
are known to be very hard. We consider that, by opening such issues, our work shows its dynamic and creative aspects and is a step towards the establishment of a body of theory and practice in e-Engineering.

We are thankful to Joby Boxall (University of Sheffield), Paul Sage (United Utilities Ltd) and Steve Hall (Yorkshire Water Ltd) for providing experimental data and feedback for our prototype system.

References


MapInfo 2008 MapInfo professional overview. See http://www.mapinfo.co.uk/location/integration.


W3C 2001 XML linking language (XLink) version 1.0. See http://www.w3.org/TR/xlink/.