

WEB BASED HYDRAULIC MODELLING

Author : *Thomas Joseph - AWT New Zealand Limited*

Presenter : *Thomas Joseph - AWT New Zealand Limited*

ABSTRACT

With the next generation of the world wide web upon us, and more and more complex computer applications being hosted in the web environment, it is inevitable that hydraulic modelling will soon have its place on the web.

This paper presents a case study that uses an open source solution base to build a robust platform to view real time monitoring data and run hydrologic and hydraulic models within a web environment. The application allows the end-user full control of hydraulic model inputs and enables them to create, delete, edit, and run “what if” scenarios under a full parent/child implementation of scenario management.

The case study is based on a water supply model however the application has been built and tested for integration with many types of hydrologic and hydraulic models and modelling software. The application can utilise base data from Google Maps, run within Google Earth, or conversely use user defined base data. The application is fully secure and only allows access to invested stakeholders or interested parties.

Using the now commonplace web hosting model, the burden of heavy computational engines is transferred to the host server and the end-user only requires a web browser and an internet connection. The web application makes it possible to organise and analyse data spatially, run scenarios, and view results on interactive maps as well as with tables and charts. The web integration allows modellers, managers, and community representatives in different locations to coordinate and share information in real time with an easy to use interface.

The end-user has access to more computing power and can immediately utilise the latest version of their system model on the server. They can also access real-time pressure, event and flow information from their network if this has been equipped with web-based monitoring.

This is the future and it is available now.

KEYWORDS

Hydraulic Modelling; Web Applications.

1 INTRODUCTION

With the onset of “Web 2.0”, the concept of giving users the free choice to interact or collaborate through a web based interface, and rapid growth in broadband coverage it is only a matter of time before engineering planning and related software products move into a web environment. Web applications provide a great media where all interested parties can be connected to “real-time” up-to-date information. By using a standard client server based model we can not only provide a single access point for all data but also shift heavy computing back onto the servers where it belongs. Open source software offers an excellent base to establish such a client server model and allows full flexibility to customise and infinitely expand without the burden and restrictions of licensed software.

In this paper I will discuss the development of a dynamic web based delivery system developed for viewing and editing hydraulic modelling associated data. The delivery system is then showcased through a case study for a water distribution model for Porirua City Council in Wellington, New Zealand. In particular the discussion will focus on the Porirua water supply model where the model was enabled to be simulated over the web and instantaneous results made available to the user. The paper will begin with a discussion around the core technology used for the web based service and finish with a discussion around the array of possibilities the technology opens for engineering planning.

2 THE TECHNOLOGY

The core technology used for building this web service was carefully selected. The core hardware/software was examined based on three primary qualities to ensure the right foundation was constructed; flexibility, expandability and speed where the most important qualities during the technology selection process. Following a detailed assessment of the currently available technology in both the proprietary and open source arenas, open source technology was deemed the best way to ensure ultimate flexibility, minimise risk of expansion costs and license issues, and provide significant scope for speed improvements as the technology improves.

The standard multiple layer architecture was used to implement the service including a data base layer, a logic layer, and presentation layer. The architecture is run from several virtual servers to ensure ultimate speed at every layer. The following sections discuss the architecture of the system and the core technology used in the service.

2.1 SYSTEM ARCHITECTURE

The system was architected with the standard three layer architecture including a data base layer, a logic layer, and presentation layer. The database layer is the foundation of the system and stores all relevant spatial and non-spatial data. Using an enterprise level data management system data can be accessed and written in a true multiuser environment with no compromises on speed or data access. The logic layer is a customised piece of software that manages all communications between the presentation layer and the data layer. The logic software dynamically feeds data into complex calculation engines. The presentation layer is the user interface which translates requests into results that are understandable by the common users.

The architecture has been designed for virtually limitless expandability and future development. Centralisation of data not only facilitates good data management but lends itself to the web architecture for easy secure access to all relevant data for any interested stakeholders. Each layer of the architecture sits on its own separate virtual server maximising the processing and memory availability and making migration to cloud computing an easy next step. The general architecture is shown in Figure 1 below.

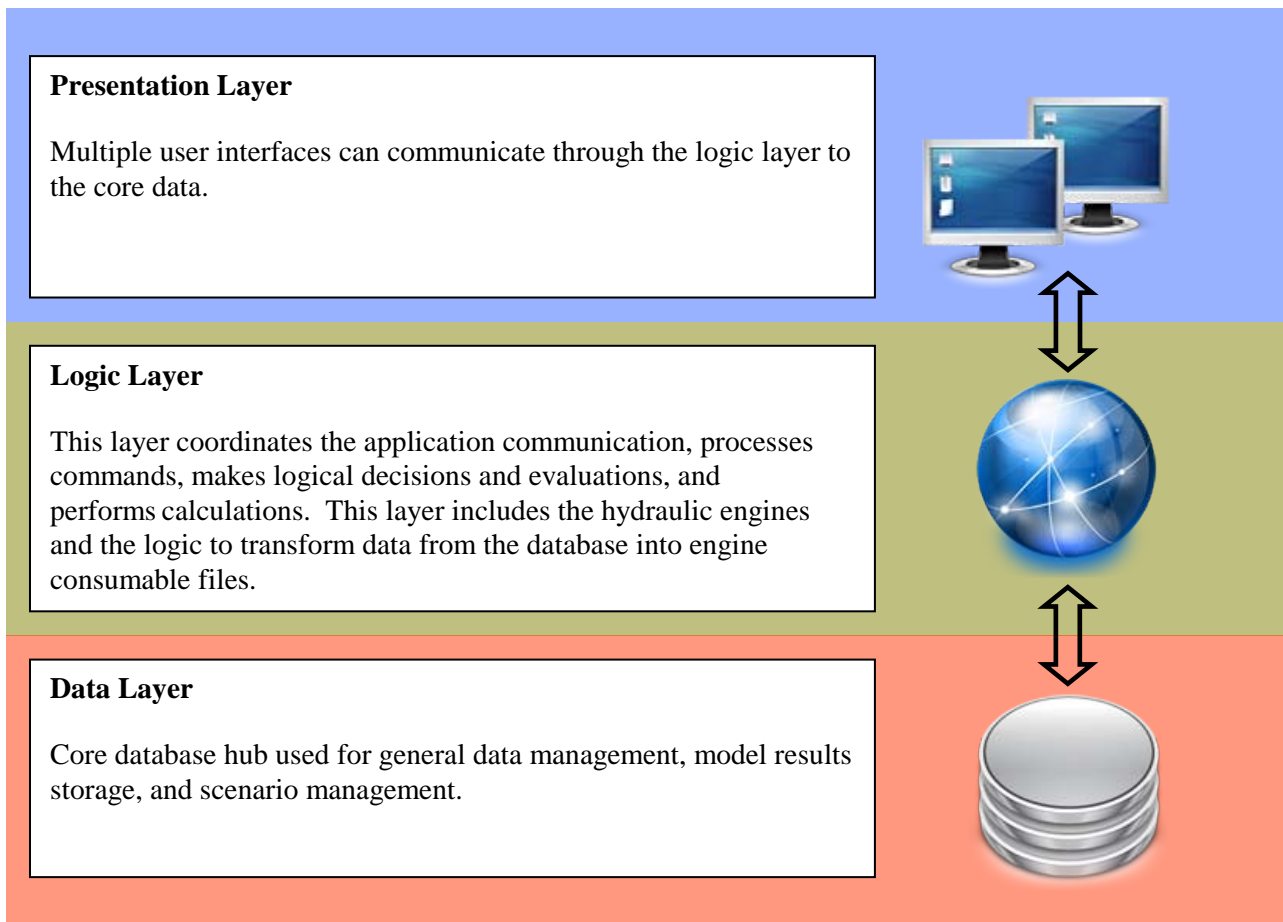


Figure 1: Three Layer Architecture Used for the Core System Architecture

2.2 TECHNOLOGY STACK

As discussed above the core software for this project was chosen carefully to ensure that the ultimate expandability and customisation was possible with minimal to no licensing limitations. The technology stack consists of multiple pieces of software within the logic and presentation tiers all running from a single core set of data from the data layer. The following section discusses the technology used in each layer.

2.2.1 DATA LAYER

A PostgreSQL database was chosen for core data storage and retrieval. We chose a Postgre database for several reasons including:

- Open source licensing provides ultimate options for expandability,
- Large user base,
- Well established within the geospatial industry as the leading open source geospatial database with the most native geospatial functions available, and
- Conforms to the Open Geospatial Consortium standards for geospatial data storage, providing data and view opportunities with several other leading geospatial software packages.

The database has been carefully structured so that it is generic enough to house any type of data, however it has been particular suited to hydraulic modelling and network planning data. Having a generic structure will allow for different modelling engines to be used in the future. The database selection process was not taken lightly, as part of the selection process we tested the performance of the Postgre database against other well know

databases including Microsoft's SQL Server and MYSQL. We found that the Postgre database outperformed the other databases in most geospatial functions and found the performance to be comparable for non-spatial data access.

2.2.2 LOGIC LAYER

The logic layer is made up of a base controller (or "engine") application which performs logic and controls communication between the user, the database, and other calculation engines and logic layers. The engine is a custom web service application written in C#. The engine is capable of communicating using both standard http and TCP/IP communications protocols allowing both internal and external communication to be handled. The logic layer also handles security, severing communication lines to the data without proper authentication.

Any number of pre-built calculation engines that perform custom calculations such as hydrological/hydraulic modelling engines, time series manipulation engines, and biological modelling engines can be managed by the engine. The engine essentially pulls data from a centralised database and feeds any calculation engine with the appropriate data in the correct format, when the calculation engine is finished with the requested calculation the engine organises the outputs into the database. The engine currently includes communication interfaces for all of the common hydraulic modelling software packages including Infoworks CS and WS, DHI MOUSE, and EPANET and EPASWMM. As discussed above the database has been architected in a generic enough format that any model can be added to the logic layer with relative ease. This structure allows for virtually an unlimited number of calculation engines (customised or proprietary) to be strung together using the same base data at the same time from any user interface.

We have also taken advantage of MapServer, another well established piece of open source software, for publishing spatial data and interactive mapping applications from the logic layer to a web presentation layer. MapServer has native support for Postgre geospatial data and provides fully dynamic geospatial publishing. As with the database software a detailed selection process was undertaken to choose an appropriate map server application. Having dynamic (always up-to-date) content was of utmost importance. MapServer provided a much more dynamic layer generator than other popular proprietary software's examined and was therefore chosen as the geospatial server of choice. A schematic diagram of the complete technology stack including the logic layer is presented in Figure 2 below.

2.2.3 PRESENTATION LAYER

The presentation layer is what the user sees and interacts with. We are currently utilising several user interfaces to interact with the data. As with the logic layer the structure is set up so that virtually an unlimited number of presentation layers can be used to interact with the data at the same time from multiple locations creating a true multiuser environment. Currently we are primarily using Google Maps and Google Earth as our geospatial presentation layer and standard web pages for non-spatial data presentation. Google Maps and Google Earth were chosen for their semi open source nature and their usability, flexibility, and affordability for any user. However more critically Google Earth and Google Maps provide an excellent core set of base data and a well documented web based communication interface (API). This communication is critical to data centralisation and creating a global multiuser environment. It is however important to note that although the Google Earth interface provides faster data access and a really cool 3D viewing environment the user does require a local copy of Google Earth on their computer. There are also some minor licensing fees involved with using Google Earth for professional purposes.

Although we are currently using Google Earth and Google Maps for our geospatial backdrop the system has been developed to easily migrate to a custom base map using only the open source MapServer if Google removes or limits their communication interface or starts advertising on the base maps.

We also have internal presentation layers through MS Access and Excel and ESRI's ArcGIS which licenses direct communication to Postgre spatial data. A schematic diagram of the complete technology stack including the presentation layer is presented in Figure 2 below.

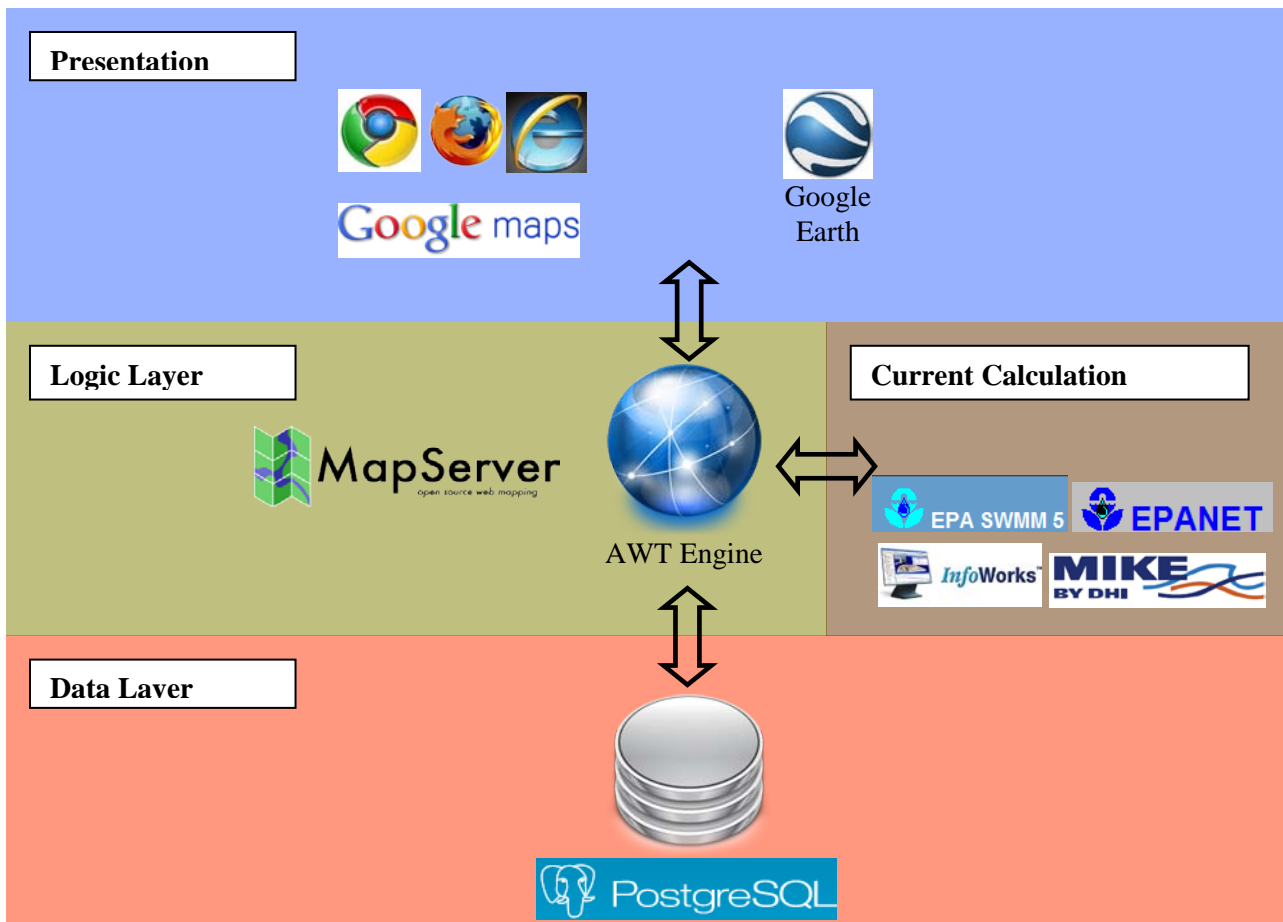


Figure 2: Technology Stack

3 THIS ALL SOUNDS COMPLICATED WHY ARE WE DOING THIS AGAIN ?

Ok all that program architecture sounds fancy and complicated but why are we doing all this. We saw the development of this architecture as the first step in presenting up-to-date dynamic results from hydraulic models to all interested users not just advanced hydraulic modellers. By providing dynamic free access to modelling results through the web not only empowers the stakeholders but results in a far superior decision making tool with continuous input from all interested parties. The easy to use interface also allows non-technical users to create and run simple “what if” scenarios and view the implications of possible solutions. We are confident that web based applications will become common place in the future. The following section outlines a few of the advantages to the developed system.

3.1 TRULY DYNAMIC PRESENTATION

The architecture of the developed system provides a truly dynamic presentation of results so that all users are viewing the same data. If changes are made by any users they are seen immediately. This creates a perfect environment for multiple users requiring the same up-to-date data. All of the data is always dynamically streamed from the engine to the requesting presentation layer. This is standard web based application methodology and is inherent to Google Maps.

Google Earth morphs this web based technology into a desktop application that provides access to static local content or dynamic streaming content. Several of the most popular commercial GIS and modelling software packages already provide simple export functions to Google Earth. This simple functionality can be used to transfer static data between users. However things change rapidly and a more dynamic solution for data transfer and sharing was required.

The network link feature within Google Earth enables users to stream data over the internet. This translates to always up-to-date spatial data.

3.2 TRUE MULTIUSER ENVIRONMENT WITH SCENARIO MANAGEMENT

The logic engine has been written as a multi-threaded application that can handle virtually an unlimited amount of users. All users see the same data and the data can be accessed from any location that has an internet connection.

A full scenario management tool has also been built into a web based presentation layer. The tool allows the user the ability to switch between scenarios and the functionality to create new and run simple “what if” scenarios. The user can display, analyse, query, share, and report results from any scenarios directly from their web browser or preferred presentation layer. Standard link and node themes are provided for the user to develop colour-coded maps based on a number of different themes including overflows, pressure, pipe capacity ratios, velocities, headloss in pipes and many more.

3.3 SECURITY DATA ACCESS

The three layer architecture offers relatively robust security. By placing the engine layer in between the presentation layer and the core data there is a level of protection above the data. All communication through the engine is verified for security and if the user is not properly authenticated the communication is stopped.

3.4 NO UPDATES REQUIRED

One of the best things about web based applications is that bug fixes, patches, and new releases are continuously happening. The user never has to re-install a new version of the application or download a patch by the very nature of the application you always have the latest version.

4 CASE STUDY

We have been working with Porirua City Council (PCC) on developing a comprehensive water network model for the entire city. The model has been developed in EPANET, an open source network modelling software available from the United States Environmental Protection Agency (EPA). We chose to model the network in EPANET so licensing issues would not cause potentially issues with web based model runs; however any water network software package could theoretically be used.

All of data required for the model was stored in the centralised database and presented to PCC through the presentation layer. At the onset of the project we sent through a single Google Earth network link file which contained an encrypted URL which streamed live up-to-date data to there instance of Google Earth through standard internet communications. As a standard outcome of the architecture a secure web page was also established so interested users without Google Earth could view the same data using Google Maps. This setup empowered the PCC project manager and as the project progressed the PCC project manager was able to see live progress on the model.

In the following section I will use this case study to demonstrate how the system works from the user perspective.

4.1 PORIRUA CITY WATER NETWORK OVERVIEW

In a typical summer day Porirua City is supplied with approximately 17.1 ML of drinking water from the Greater Wellington Regional Council (GWRC). The demand is serviced using 18 reservoirs servicing 9 demand zones. The entire network has been modelled which consist of approximately 340km of pipes.

4.2 PRESENTATION LAYER

The primary presentation layer used by PCC is the Google Earth interface. This interface was chosen by PCC primarily due to the 3D viewing capabilities. The basic presentation layer is made up of three primary features the map and spatial results presentation, results summary and base data table viewing, and scenario management and model simulations.

4.2.1 MAP AND RESULTS PRESENTATION

By using Google Earth and Google Maps base layer information, including roads and aerial imagery, the base data is served by Google and does not have to be stored in the data layer or handled by the engine. The network information including pipes, nodes, pumps, and reservoirs is dynamically served from though a web service to Google Earth via network links. We have significantly optimised the network link load time by using Google Earth regions so only the area you are zoomed to is streamed and loaded. All the available result themed presentation layers are included in the standard places menu and the user can turn layers on and off. The find bar can be used to search for asset names or addresses. A dynamic legend is created using MapServer and fixed to the bottom left corner of the map using a Google Earth ground overlay. All of the standard Google Earth features including street view are available for use with the interface. Figure 3 below shows an example of the maps presentation layer.

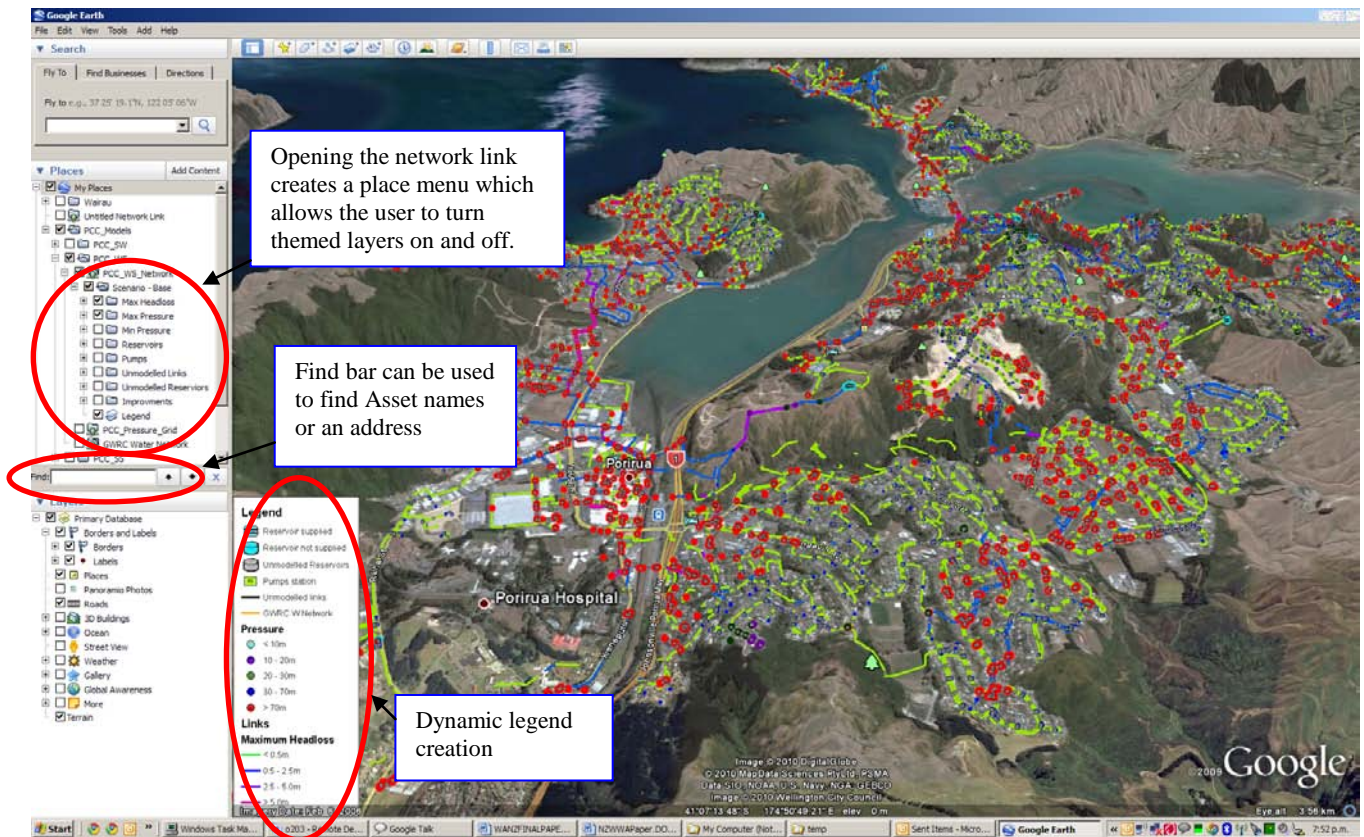


Figure 3: Google Earth Map Presentation Layer

4.2.2 VIEWING BASE MODEL DATA AND RESULTS IN THE DETAILS VIEW

All network elements on the map are clickable and have links to a web form where live model data and results are tabulated and can be viewed. The details view displays all of the data used in the model for that particular element and associated source for data. This empowers the user with the source information used to build the model so appropriate limitations can be understood. Within the details view the user can also look at any validation errors at the element or any notes that have been logged at the element. The results for all of the

scenarios that have been run are also shown in the details view. This enables the user to easily compare results at a particular element for multiple scenarios. Figure 4 below shows an example of details table views.

Validation errors and notes can also be viewed.

Base model data is displayed with data source so areas of data estimation can be identified

Results for all scenarios are shown so result comparison can easily be completed

AssetName	Diameter	Invert Level	Lid Level	Additional Demand (l/s)	Source
85585	1.05	62.005			DataSetID N_Dia_DG N_IL_DEM
ShortID	Max Demand (l/s)	Peak Pressure (m)	Min Pressure (m)		
Base	0.02240798	42.28638	42.27995		
WhitbyCoastalEstates	0.02240798	42.28638	42.27995		

Figure 4: Accessing the Details View

4.2.3 CREATE SCENARIOS AND RUNNING THE MODEL

The details view can also be used for scenario management and running model simulations. From the scenarios tab in the details view the active scenario can be changed, created, and if you are the owner of the scenario it can be deleted. If the user owns the active scenario details can be edited and the model re-run. The run scenario button will submit the scenario to run on the server. If the scenario is changed or model run the map layer will dynamically be updated using the Google Earth refresh command. Figure 5 below shows a detail of the scenarios tab.

The scenario tree can be used to switch scenarios, add scenarios, or delete scenarios

Edit demand at the node to see how pressure changes in the area

Base Scenario: WhitbyCoastalEstates						
AssetName	Diameter	Invert Level	Lid Level	Additional Demand (l/s)	Source	
85585	1.05	62.005			DataSetID N_Dia_DG N_IL_DEM	

Child Scenario: ttt - details are inherited from the base scenario and can be edited						
	AssetName	Diameter	Invert Level	Lid Level	Additional Demand (l/s)	Source
Save	85585	1.05	62.005		55	DataSetID N_Dia_DG N_IL_DEM
Cancel						

	ShortID	Max Demand (l/s)	Peak Pressure (m)	Min Pressure (m)
Base		0.02240798	42.28638	42.27995
WhitbyCoastalEstates		0.02240798	42.28638	42.27995

Figure 5: Scenarios View

4.3 SOUNDS GOOD, IS IT USEFUL

PCC has been using the model presentation interface for over a one year now and have found the data access very useful for in network planning and operations. We have extended the model interface to the stormwater and wastewater models and now all three water networks available through the interface. We have found the interface to be really good for picking up issues the model and areas where predictions are not quite right. The interface is a great way to keep model content dynamic as the system changes. The success of these projects has prompted more investment in to this technology to expand into housing other dynamic planning data such as reports and drawings, flow and environmental monitoring data, CCTV records, site photos and much more.

5 LOOKING FORWARD

Although a solid base architecture has been built we are currently just scratching the surface of this technology. We are constantly added features to the presentation layer and developing more links to other calculation engines. We are also moving towards opening the interface to the user community to grow the project faster with input from all. We feel that web based software is most definitely the future. Web based services are made for cloud computing which lowers IT and software costs and speeds up the user experience. It won't be long before the user will only require the most modest of PC's with a web browser and an internet connection to run the most complicated models.

6 CONCLUSIONS

In this paper we discussed a web based architecture for delivering hydraulic modelling data and results to a multiuser environment. By providing dynamic access to model build data, model scenarios, and modelling results not only empowers the stakeholders but results in a far superior decision making tool with continuous input from all interested parties. The easy to use interface also allows non-technical users to create and run simple "what if" scenarios and view the implications of possible solutions. The web integration allows modellers, managers, and community representatives in different locations to coordinate and share information in real time with an easy to use interface. The end-user has access to more computing power and can immediately utilise the latest version of their system model on the server. We are confident that this type of application will become common place in the future.

ACKNOWLEDGEMENTS

I would just like to say thanks to Porirua City Council and in particular Desmond Scrimgeour for giving this new technology a try and providing us with valuable feed back on the interface performance and other related issues.