WebSense: A Lightweight and Configurable Application for Publishing Sensor Network Data

Rachel Cardell-Oliver #1, Christof Huebner *2, Miriam Foeller-Nord +3

School of Computer Science & Software Engineering,
The University of Western Australia, Perth, Australia
[1] rachel.cardell-oliver@uwa.edu.au

* *Institute for Industrial Data Processing and Communication,*
University of Applied Sciences Mannheim, Mannheim, Germany
[2] c.huebner@hs-mannheim.de

+ *Institute for Embedded and Mobile Computing,*
University of Applied Sciences Mannheim, Mannheim, Germany
[3] m.foeller@hs-mannheim.de

***Abstract*—Wireless sensor networks generate volumes of scientific observations. However, gathered data is typically not published for use by other scientists. This paper analyses the reasons that so little sensor network data is published online in a usable form, and identifies gaps in currently available publication systems. To address these gaps we present WebSense, a lightweight and configurable web-based tool for sensor network researchers to publish their collected data. The design goals and implementation of WebSense are described and the system is evaluated against the goals.**

Keywords: Wireless Sensor Network, Observations Metadata, Data Visualisation

I. Introduction

Sensor network research projects have generated volumes of observation data for application domains including agriculture, environmental science, engineering and health care. However, very little of the data collected in these projects has been made available in the public domain for other scientists. This problem is known colloquially as the "data graveyard problem". The problem is significant for sensor network researchers, not least because the "impact of sensor networks for habitat and environmental monitoring will be measured by their ability to enable new applications and produce new results otherwise too difficult to realize." [12]

This paper investigates why data graveyards occur, and then proposes a lightweight and configurable web-based publishing tool called WebSense to address the problem. WebSense uses open-source technologies to build a simple to install and use, configurable application for publishing sensor network data. WebSense fills a gap in existing solutions for publishing sensor network data by supporting sensor network researchers who wish to maintain control and ownership of their data and to publish it in a usable form with the minimum of fuss. WebSense has been evaluated using case studies for a variety of different sensor network deployments.

Our goal is to present usable, high quality data whilst minimising the time and effort required for its preparation. The novel features of WebSense are: 1. supports existing models for both physical and domain properties of sensor network data and metadata; 2. the developer maintains ownership and control of their data; 3. lightweight protocols for publication and maintenance of data; and 4. a configurable and modular presentation layer that can be readily extended.

II. Background and Related Work

From the sensor network researcher's perspective, the main barrier for publication of sensor network data is the effort required to publish data in a form that is useful to application domain scientists. There are three design areas to be considered:

1) **Data and Metadata Models:** Sensor network data comprises volumes of observations made by sensors. Metadata about those observations is critical to enable users to interpret and use that data. What metadata should be provided and for which user audiences?
2) **Publication Protocols:** How is the process of populating and maintaining the published data managed? Does the publisher maintain ownership and control of their data or is this trusted to a third party? Sensor network data is typically noisy with many errors; when publishing the data, what how is data quality enforced?
3) **Presentation Layer:** What visualisation and analysis capabilities are required by users? Can the user community modify and contribute their own presentation layer functions, or are these pre-set?

Design decisions for the publication of sensor network data concern data and metadata models, publication protocols and the presentation layer. A number of different approaches to these problems have been developed, most of which are based on either web services or on distributed database architectures. We now review existing publication techniques in terms of the design choices.

A. Web Services

Web service architectures support the publication of and shared access to information over the internet. Data is formatted using a common markup language so that it can be discovered and shared by any web service application. Anyone can develop service applications that discover and present relevant data, and so a rich range of presentation services can be provided by contributions from users around the world. For web services to work, there needs to be a common language for annotating observations. The Open Geospatial Consortium (OGC) is developing syntactic standards for specifying observations [5] and sensors [6]. Current work on semantic sensor networks and semantic sensor webs offer semantic annotations to describe the meaning of sensor network data [4].

Applications that support the web services approach for sensor networks include SensorWeb and SensorBase. SensorWeb users go register sensors on a web page and start streaming data. They can use applications such as SensorMap to view that data and data from other subscribers [8]. SensorBase users blog their sensor observations to a shared server and the system provides some presentation services for viewing that data [3].

Web service architectures are loosely coupled systems. Any data producer can publish their annotated data on the web and any users can publish applications that search for and present that information. However, there are some disadvantages to this approach. Users must agree on standards for annotation. Although publication of data on the web makes that data public, it does not ensure that the data is accessible or usable. And suitable web services for visualisation and analysis of that data must be written or discovered by sensor network data users. From the sensor network researchers' point of view, suitable services may be available, but if not then changing the representation, quality control or visualisation of data is a complex and time consuming process.

B. Distributed Databases

Several scientific communities have supported large scale, cooperative projects to agree a common data and metadata model for a shared, distributed data base, together with a library of applications for accessing that data. Users can overcome the problems of different data formats and meanings and lack of applications by converting their data into the commonly agreed format and publishing it in a distributed database system. The Observations Data Model, developed for hydrological modelling [9] is one example of this approach. Australian Water Resources Information System [1] another framework under development that aims to provide uniform access to data from many different providers. The main advantage of the distributed database approach is that specific scientific communities can develop a rich data model for their particular domain (e.g. hydrology), with quality control standards. However, models for specific domains will be unsuited for other domains, which is inconvenient for sensor network researchers whose deployments may come from quite different domains. Additionally, distributed database models can not easily be re-configured by their users, since visualisation applications as well as the underlying database model are strictly controlled by the developers.

Another version of this approach is to develop bespoke web-based applications for managing sensor network data. CSIRO's Sensornets web application [18] is an example Such applications are focus on highly professional presentation of data, and efficient and reliable data storage. However, they are not designed to be re-configured or extended by users. And they are costly to purchase and set up.

C. Research Gaps

Several different approaches exists for publishing sensor network data: data models range from the simple to the highly complex, publication protocols range from loosely controlled protocols that push data to the web to strictly controlled data quality processes. Presentation layer functions can be either loosely coupled or tightly coupled with the application.

WebSense fills a gap within this design space: it is aimed at developers who are sensor network scientists wishing to make their data available with relatively low effort, but also to be able to use that data for quality science in application domains. The application should be able to be installed, populated and running in a few hours. The application must also offer the flexibility of a highly modular and configurable design so that developers who are not software engineering experts are able to configure their version of the application for different project needs.

In the following sections we describe the design choices made for WebSense in the areas of data and metadata models, publication protocols and the presentation layer. A case study implementation is presented and evaluated against our design goals.

III. Data and Metadata Models

At its simplest a sensor network observation is defined by its what (value was observed), where (place) and when (time). However, a collection of such observations on its own is of little use without *metadata* for its interpretation. Metadata is additional information about the phenomena observed and the observation process that is needed to interpret observations. Specifying sufficient metadata is an important aspect of publishing sensor network data.

The literature offers a wealth of models for observation data and metadata: domain observation models [9], OGC syntactic standards [5], [6], and semantic web ontologies[4]. WebSense draws from each of these models. Both the user perspective (properties for interpreting the domain) and the developer perspective (physical sensor network properties) are addressed.

A data model is an abstraction that can be implemented in different ways. For example, in XML files as semantic web data or as tables in a conventional database. WebSense has a database implementation in the open-source PostgreSQL system [16]. Table III summarises the data and metadata model implemented in WebSense. Developers are free to extend of modify this data model, since WebSense allows the developer to maintain control over their database. The disadvantage of this approach is limited interoperability, which could be addressed in future by providing export and import from the WebSense model to some existing sensor web standard(s).

IV. Publication Protocols

Having designed a data model, the next problem is populating that model: How is data created, entered and maintained? And how is data quality maintained?

The flow control for the WebSense publication process is shown in Figure 1. Data is produced by deployed sensor networks and then transformed by shell script programs into an SQL template for inputting observations, together with metadata about the sensor network also encoded in SQL templates. In the second phase, the SQL templates are executed to populate the database. The database is viewed via a web server that provides data consumers with different views of the data.

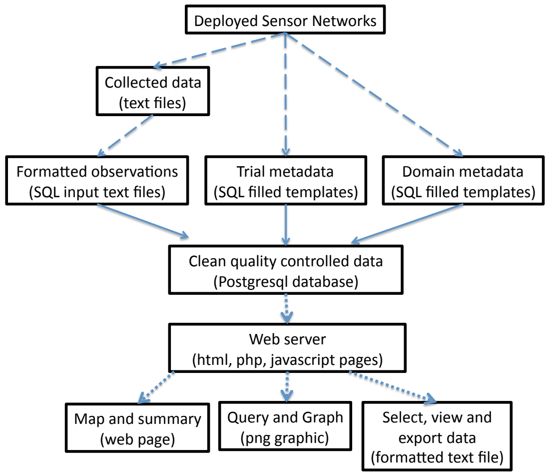

Fig. 1. **WebSense control flow from data producers to consumers. Data transformation software is implemented as shell scripts (dashed arrows), postgre SQL (solid arrows) and PHP and Javascript (dotted arrows).**

WebSense data is formatted, uploaded and maintained in a PostgreSQL database [16]. Example SQL templates for sensor network trials are provided for users to modify and upload. The sensor network trial template is 60 lines long, and the data measurements template 15 lines, so they are easy to understand and modify.

Data quality is maintained in WebSense using three techniques: controlled vocabularies, data attribute constraints, and database rules.

Controlled vocabularies such as ODM measurement methods (see Table 2 [9]) are used to maintain the integrity of selected attributes. Controlled vocabularies are implemented in WebSense using PostgreSQL's enumerated types. For controlled attributes the user can only enter observations from the particular enumerated type.

Data attribute constraints can be used to enforce non-null values, specified default values, and primary and foreign keys in database tables. Such constraints maintain the relationships between tables (typically identifier references) and control the validity of data entered into a single table.

Data attribute constraints alone can not maintain necessary conditions between tables. For example, to enforce that a sensor reading lies within the allowed range for that sensor we need a different approach. SQL rules are written to check for non-conforming measurements and then to delete non-conforming data from the database. Such rules are applied each time data is uploaded, to ensure the integrity of data in the database. The same approach can be used to identify and report alarm conditions, which will be implemented in future versions.

V. Presentation Layer

WebSense separates the roles of readers (users) and writers (owners) of its data. The back-end PostgreSQL database can be modified by developers but is only accessible for reading via the presentation layer of the WebSense application. The main challenge here is flexibility: providing an extensible presentation layer that users can configure for their own needs whilst not compromising the integrity of the database. A typical list of capabilities for an environmental visualisation tool includes mapping, searching, downloading, plotting, dashboard, tabulation and analysis applications [1].

WebSense is a mash up that utilises the open source web tools GoogleMaps [7], OpenStreetMaps [14], JpGraph [10]. WebSense achieves flexibility in accessing these tools through the high level languages: PHP [15], Javascript and HTML. WebSense provides presentation capabilities as independent modules. The system currently supports visualisation by map, search, download, plot, tabulate. Further interfaces will be addressed in future releases. For example, interpretative analysis of data could be provided by SenseWeb-style "transformer" applications that analyse the raw data to extract high level semantic information [8].

A. Maps

Three options are currently available for displaying a summary map of a sensor network. GoogleMaps service provides dynamic maps for any longitude and latitude location together with overlays [7]. OpenStreetMaps provides similar functionality to GoogleMaps but its maps are entirely open source and can be edited by users [14]. Therefore, users can choose to add relevant details for a sensor trial, such as vegetation or contours. WebSense also allows the user to provide their own image where a geographical map is not appropriate. For example, a map showing sensor placement in the interior of a building, or the placement of underground soil moisture sensors or of sap flow sensors in tall trees.

B. Browsing Trial Summaries

When a user first accesses a sensor network data archive they need a high level view of the data. WebSense provides a summary of each trial or observatory using a map, plus summary statistics (position, dates, number of nodes, number

Entity	Attributes
observatory	id, description, latitude, longitude
trial	id, observatory_id, description, latitude, longitude, timezone, sourcecontact, mappreference, mapimage
platform	id, description, latitude, longitude, serialnumber, trial_id
sensor	id, description, latitude, longitude, elevation, serialnumber, sensordevice_id, node_id
subsensor	id, description, instrument_id, phenomena_id
measurement	time, value, subsensor_id
phenomena	id, description, unit, minimum, maximum, comment
sensordevice	id, description, measurement_method, calibration, datasheetlink
history	trial_id, time, comment

TABLE I
WEBSENSE DATA AND METADATA MODEL

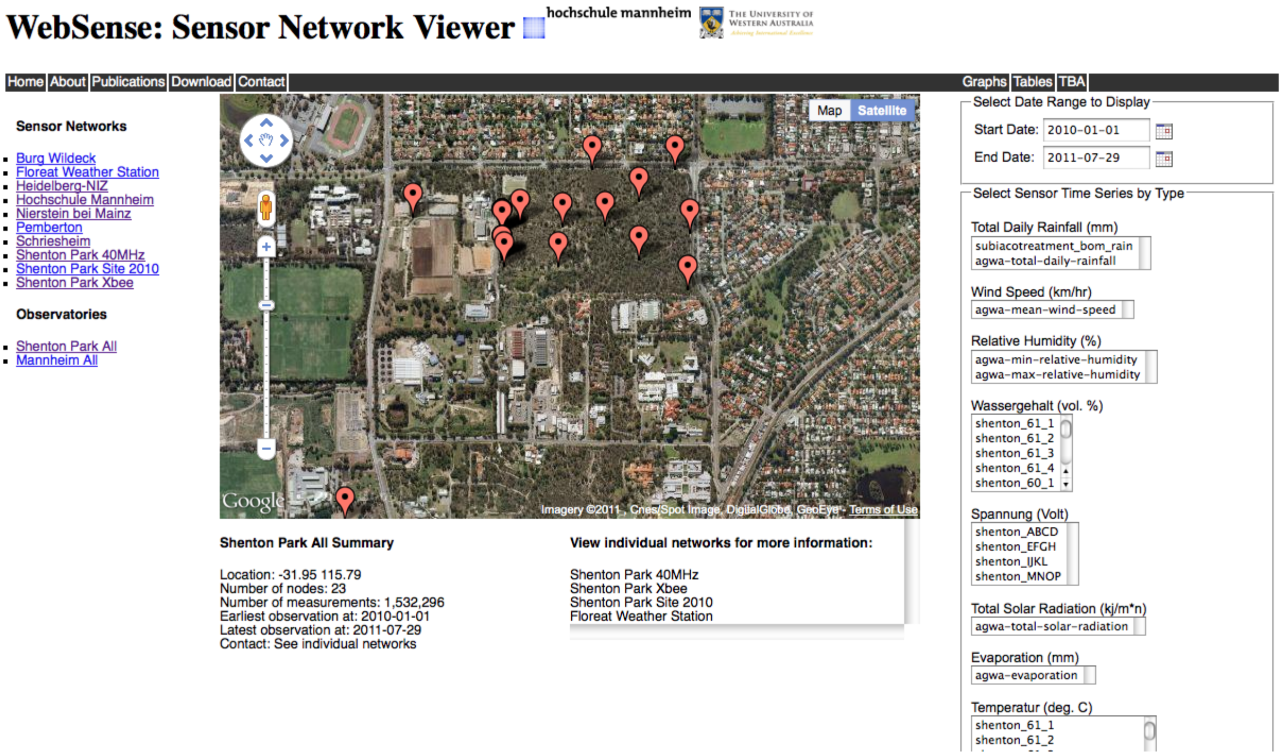

Fig. 2. WebSense web page view

of observations). At the same time when a trial is selected the query window gives a list of phenomena and the temporal extent and total number observations. The trial history can also be browsed. This information enables users to browse WebSense data at a high level, and then select the data they want to analyse in more detail.

C. Queries, Graphs, Export and Tables

A query form allows users to search and select WebSense data. Users can pose SQL queries for a given time range and selection of sensors. The results of a query can be viewed either as a time series graph as shown in Figure 3 or exported as a csv file for further analysis.

Users can also browse the metadata in tabular form. Metadata available in this form includes the real world observation phenomena supported by WebSense and details of physical sensor devices.

VI. IMPLEMENTATION AND EVALUATION

In this section we evaluate the WebSense system against the requirements of a rich data model, lightweight publication protocol and configurable presentation layer.

1) Flexible Data Model: Data and metadata models for WebSense are based on existing semantic models for sensor network data. These models cover both the developers' view of the physical organisation of sensors and nodes in a sensor network and the users' view of observed phenomena and their interpretation. The model has proved sufficiently flexible for various applications in environmental monitoring and a building monitoring application in development. Although most of the data in our current implementation is from our own network deployments, data has also been imported from

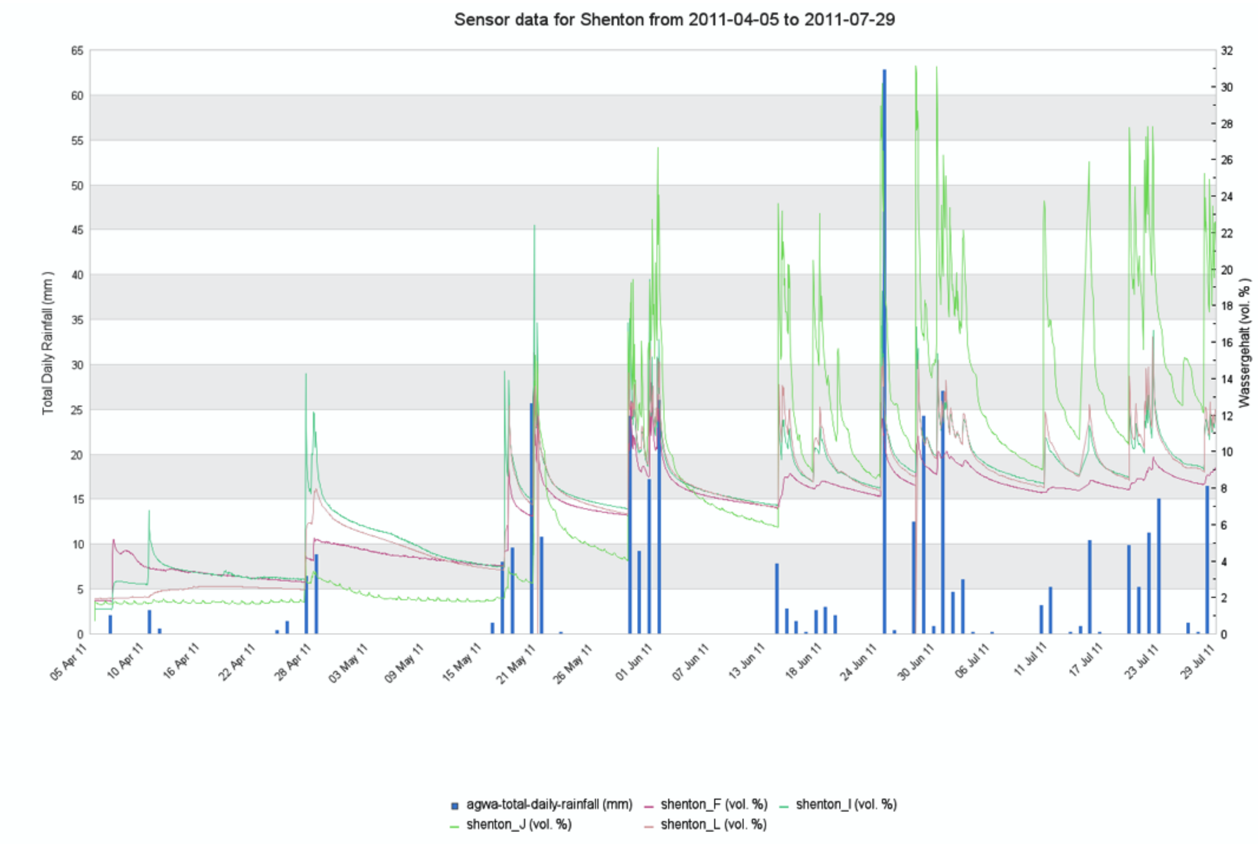

Fig. 3. WebSense Graph View: Changes to soil moisture (wassergehalt) in response to daily rainfall (bars) during Winter.

external sources such as Department of Agriculture weather stations [13].

2) Understandable Code Base: The WebSense application consists of 15 PHP files, with a total of just over 700 lines of code for the application. This can be contrasted with thousands of lines of code for equivalent applications in Java [11] or J# [17]. The small code base of WebSense can be understood and so extended or modified by users who are not software engineering experts.

3) Minimum Effort Data Entry: To add a new trial to the database, users complete and load an SQL template given as a text file of ≈ 60 lines. Shell scripts are used to upload raw data, convert it into database input format, and upload it to the database, using database checks to ensure the quality of saved data. The data upload process can be fully automated with regular scheduled updates.

4) Modular and Extensible Application Layer: The presentation layer is implemented as a collection of PHP web pages that access the WebSense database, and other applications such as maps and time-series graphs. New presentation layer components can be implemented by adding, removing or modifying these PHP templates as needed.

5) Support Software and Installation: WebSense requires libraries for JPgraph [10], Postgresql and the Apache server for PHP. Bundled Apache/Postgres downloads are readily available for Windows (WAPP), MacOS (MAPP) and Linux (LAPP) [2], [16]. A new user can populate and install WebSense for their own applications in a few hours.

6) Scalability: Our example database contains 3.6 million observations (3,612,148) from 275 deployed sensors (890 subsensors) in 10 networks. Whilst Postgresql can manage very large data sets, some care is needed in the application layer. For example, in the graphing application very large time series must be filtered to a representative set of observations so that the time for display is reasonable. Currently up to 2500 observations are selected to represent each time series. Multiple axes are used to display different phenomena on the same graph: for example rainfall and soil moisture from any number of subsensors.

VII. Conclusion

The goal of WebSense is to support sensor networks researchers in the presentation of usable, high quality data whilst minimising the time and effort required for its preparation. We have presented the design of WebSense and showed how it complements existing community data bases and semantic web services. The novel contributions of WebSense are a rich ontology suited to sensor networks; lightweight publication protocols but with control of data quality; and a configurable and modular presentation layer ready for extension.

There are several directions for continuing this work. There are many possible presentation layer options such as a dashboard for current network status, spatial heat map graphs or animations, new types of interactive maps, tabular output of metadata, and data mining applications. Alarm conditions can be checked and reported as results are uploaded to the database. Closer integration of our data metadata models with evolving semantic web standards would support export and import of data sources from the sensor web.

ACKNOWLEDGEMENTS

This work was partially supported by a grant from the Go8-DAAD Australia-Germany Joint Research Co-Operation Scheme.

REFERENCES

[1] R.M. Argent, B.G. Anderson, D.G. Barratt, and T. Robotis. A Water Resources Information System for Australia. In *18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009*, 2009.

[2] BitNami: WAPPStack. http://bitnami.org/stack/wappstack, Accessed July 2011.

[3] Kevin Chang, Nathan Yau, Mark Hansen, and Deborah Estrin. SensorBase.org - A Centralized Repository to Slog Sensor Network ata. http://escholarship.org/uc/item/4dt82690, 2006.

[4] M. Compton, C. Henson, L. Lefort, H. Neuhaus, and A. Sheth. A survey of the semantic specification of sensors. In *Proceedings of the 2nd Int. Workshop on Semantic Sensor Networks (SSN09), Washington DC, USA, October 26, 2009*, 2009.

[5] Open Geospatial Consortium. Observations and measurements. http://www.opengeospatial.org/standards/om.

[6] Open Geospatial Consortium. SensorML: sensor modelling language. http://www.opengeospatial.org/standards/sensorml.

[7] Google Maps API Family Google Code. http://code.google.com/apis/maps/, Accessed July 2011.

[8] W.I. Grosky, A. Kansal, S. Nath, J. Liu, and F. Zhao. SenseWeb: An Infrastructure for Shared Sensing. *Multimedia, IEEE*, 14(4):8 –13, oct.-dec. 2007.

[9] Jeffery S. Horsburgh, David G. Tarboton andDavid R. Maidment, and Ilya Zaslavsky. A relational model for environmental and water resources data. *Water Resources Research*, (W05406), 2008.

[10] JpGraph - Most powerful PHP-driven charts. http://jpgraph.net/, Accessed July 2011.

[11] Moritz Lenz. WebSense application, Hochschule Mannheim. http://141.19.78.216:5003/WebSense/.

[12] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proc. First ACM International Workshop on Wireless Sensor Networks and Applications, September 2002, Atlanta, Georgia, USA*, 2002.

[13] Department of Agriculture and Food Western Australia. Floreat Park Weather Station. http://agspsrv34.agric.wa.gov.au/climate/livedata/flwebpag.htm.

[14] OpenStreetMap. http://www.openstreetmap.org/, Accessed July 2011.

[15] PHP: Hypertext Preprocessor. http://www.php.net, Accessed July 2011.

[16] PostgreSQL - The world's most advanced open source database. http://www.postgresql.org/, Accessed July 2011.

[17] Catherine Rye. Development of a web-based interface for analysis of environmental data from a wireless sensor networks, October 2005. Honours Thesis, School of Environmental Engineering, University of Western Australia.

[18] Sensornets CSIRO. http://150.229.98.68/deployments/, Accessed July 2011.