Handling and Publishing Wireless Sensor Network Data: a hands-on experiment

Ivo K. Koga[1], Claudia Bauzer Medeiros[1], Omar Branquinho[2]

[1]Institute of Computing – State University of Campinas (UNICAMP)
Caixa Postal 6176 – 13084-971 – Campinas – SP – Brazil

[2]PUC – Campinas
13086-900 – Campinas – SP – Brazil

{koga, cmbm}@ic.unicamp.br, omar.branquinho@gmail.com

Abstract. *eScience research, in computer science, concerns the development of tools, models and techniques to help scientists from other domains to develop their own research. One problem which is common to all is concerned with management of heterogeneous data offering multiple interaction possibilities. This paper presents a proposal to help solve this problem, tailored to wireless sensor data – an important data source in eScience. This proposal is illustrated with a case study.*

1. Introduction

Wireless Sensor Networks (WSN) [Akyildiz et al. 2002] are a special kind of ad hoc network, composed of a huge amount of small nodes with low processing capacity, limited power source, high mobility and high probability of failures (communication, power, and/or node failures) than other kinds of networks. Nodes potentially have different types and functionalities and monitor a large scale of physical and environmental variables (e.g. temperature, humidity, heat).

WSNs allow the acquisition of data in difficult or even impossible spatial and temporal resolutions and scales. The sensors can be more intimately connected with the observed phenomena, being kept active during a long time. For example, they allow detection of an animal triggering an action, or the measurement of temperature of a volcano.

The possibility of monitoring many phenomena, in many temporal and spatial scales, can produce a large volume of heterogeneous data. Heterogeneity and volume of data, combined with heterogeneity in user requirements, pose many problems. The storage, retrieval and visualization of data in this kind of setting is a challenge which is associated with one of the grand challenges in computer science defined by the Brazilian Computer Society (SBC) – Management of large multimedia data volumes [Medeiros 2008].

The goal of this work is to contribute towards solving this challenge, by proposing a practical way of storing and publishing sensor data, making possible the extraction of information by different types of users. Each kind of user profile can determine their special needs, defining what they want from the available data allowing the extraction of relevant information.

This work is related with ongoing research on the management of sensor data in eScience – in particular, for biodiversity and environmental studies. We present our proposal by means of a case study of management of environmental data in an application related to agriculture.

The rest of this paper is organized in the following sections: a brief overview of WSN data management (section 2), section 3 presents our approach, section 4 discusses our case study and finally section 5 concludes the paper.

2. Related work

Different types of systems are being proposed and deployed to support the scientist's work in many research areas. Biodiversity systems are an example of this type of system to support the work of biologists. Examples are studies in biodiversity, ecology or environment monitoring. On closed environments, sensor networks are being used in scientific studies concerning health (e.g. patient monitoring) or chemistry (experiment monitoring). See [Hey et al. 2009].

There are countless initiatives concerning WSN data management, that range from network configuration and energy management to data processing and publication [Nakamura et al. 2007, Huang et al. 2007, Zhao and Gurusamy 2008, Pantazis et al. 2009]. This paper is concerned with solutions that support the latter stage – i.e., once data are collected, how to provide flexible mechanisms to forward data to be processed and published, hiding low-level details. Our choice of related work reflects this, concentrating on their architectures for sensor data management and publication.

Many solutions were proposed to overcome the heterogeneity problem of sensor data management. Chu et. al. [Chu et al. 2006] created an architecture called NICTA Open Sensor Web Architecture (NOSA) which combines a Service Oriented Architecture (SOA) and WSNs using the services specified in the Sensor Web Enablement (SWE) [OGC 2010] from the OpenGIS Consortium (OGC). Figure 1 shows NOSA and its components. There are 4 layers: Sensor Fabric, Application Services, Application Development and Applications. The first layer deals with sensors and their emulation/simulation, the second is composed by services that support network management, the third provides the APIs, tools and configuration and the last has the applications that use the sensor data.

In NOSA, sensor data are always processed by entities external to the sensor network. This can be an advantage in scenarios where the deployment and maintenance of the sensors are easy. These scenarios consider that sensors will only sense and send data, without any processing, which consumes more power (most expensive activity in face of power consumption) [Akyildiz et al. 2002]. However, in specific scenarios, it could be more appropriate to use WSN pre-processing capacity before actually sending data. Another disadvantage is that one application that wants to use the same implemented request code from another application doesn't have the ability to do it in a friendly way.

Pastorello Jr [Pastorello Jr et al. 2007] dealt with the problem of production and management of WSN data through a framework that uses software components called Digital Content Components (DCC) [Santanchè and Medeiros 2007] and scientific workflows to provide management facilities and easy access to the sensor data. Unlike NOSA, this work does not consider the OGC standards for WSNs. However, it has some advantages such as the flexibility, letting open the possibilities for development of new DCCs for access and management of sensors, regardless of the implementation and technology of the sensors.

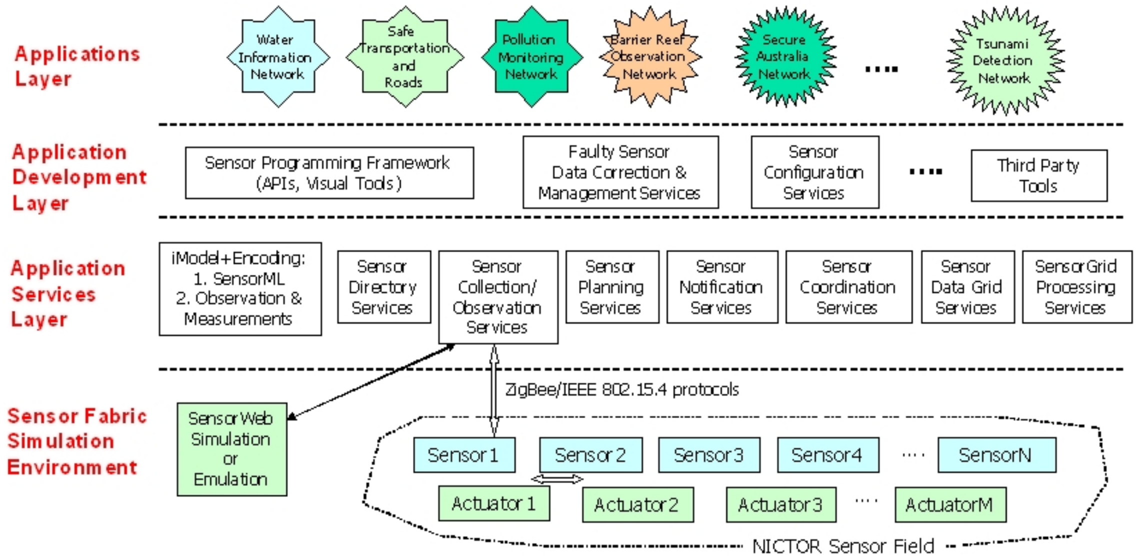

Figure 1. NICTA Open Sensor Web Architecture. Source: [Chu et al. 2006]

Global Sensor Networks (GSN) [Aberer et al. 2007] is a platform developed in Java that provides an infrastructure for the integration of technologies of heterogeneous sensor networks using a set of abstractions and XML. GSN has the advantage of facilitating the WSNs deployment when it hides its implementation details, but it also hides platform specific parameters that can make each deployment flexible.

The problems faced by these proposals range from a micro perspective (a large amount of sensors in a single network) to a macro one (between WSNs and between them and the Web). Pastorello Jr proposed DCCs and Workflows to deal with the heterogeneity problem. GSN was proposed as an infrastructure to overcome some deployment problems using XML and abstractions implemented in Java. NOSA encapsulates the operations in a software layer that uses the SWE standards and grid computing to provide a middleware that provides services that overrides sensors' implementation complexity.

In these and other efforts, the idea is to provide several layers of isolation between the sensor networks and the users. Then, one can customize and develop each layer and concentrate on the solution of a few problems at a time. As will be seen in the next section, our proposal combines features from the reviewed papers.

3. Proposed Solution

The solution used in our work is based on two aspects:

- Web services – to provide interoperability between applications, WSNs and data servers;
- Components – to support reuse and loose coupling.

Figure 2 gives a high level view of our proposal. It has three main components (or layers): WSNs (on the left), data servers (on the right), and user applications. Data communication among components is supported by Web Services. Specific functionalities are implemented by software components. Each WSN has a client responsible for running a Data Load service and sending the raw data to the server.

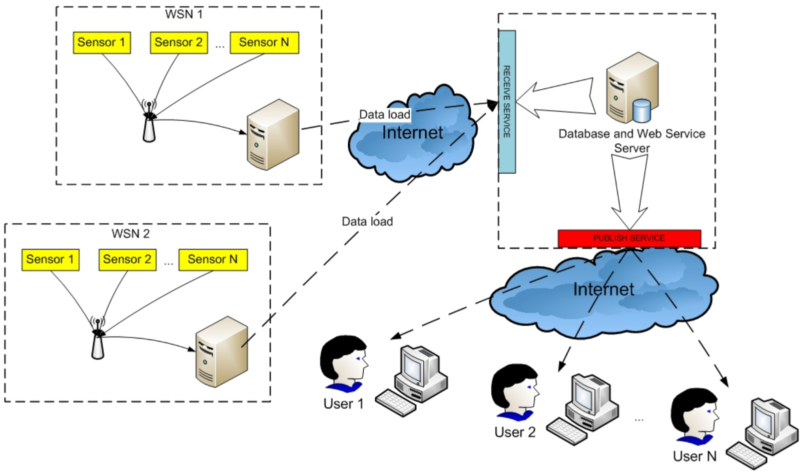

Figure 2. Architecture of the solution.

The data server implements two Web Services: a Receive service and a Publication service. The Receive service formats the data received from Data Load into standard tuples, and stores them in a database. The Publication service implements basic methods that execute SQL queries on the database.

User requests are treated the following way. Distinct query parameters and visualization requests are implemented as components that invoke the Publication service. Storage and visualization thus follow two independent pipelines. In the “push” pipeline, Data Load pushes data into the Receive service. In the “pull” pipeline, components request data from the Publication service.

This solution has the following advantages. First, it takes advantage of web services to provide access interoperability. Second, since it is based on components, it is extensible – for instance, components can also be developed at the client (sensor network) side to preprocess the data, but also (as in our case) at the server side to integrate and customize data according to distinct application requirements. Also, different servers can be installed to support, for instance, distinct needs or to integrate data from different networks.

The use of software components makes possible the development of user-specific components to access and visualize WSN data. This is shown in Figure 3 where we have distinct components for accessing and visualizing data, separating this in a Model-View-Controller (MVC) [Krasner and Pope 1988] pattern. This makes possible for one access component to be used by many visualization components and also one visualization component make use of many access components.

In order to provide a first prototype for visualization of sensor data, we extended the FLAVOR framework [Koga et al. 2007]. FLAVOR was developed to support flexible design and construction of software components to visualize measurements of network traffic. We point out that such measurements can be treated as time series – and thus FLAVOR could be adapted to deal with our sensor measurements. As new requirements for other functionalities for accessing and visualizing WSN data appear, certainly FLAVOR can be progressively extended.

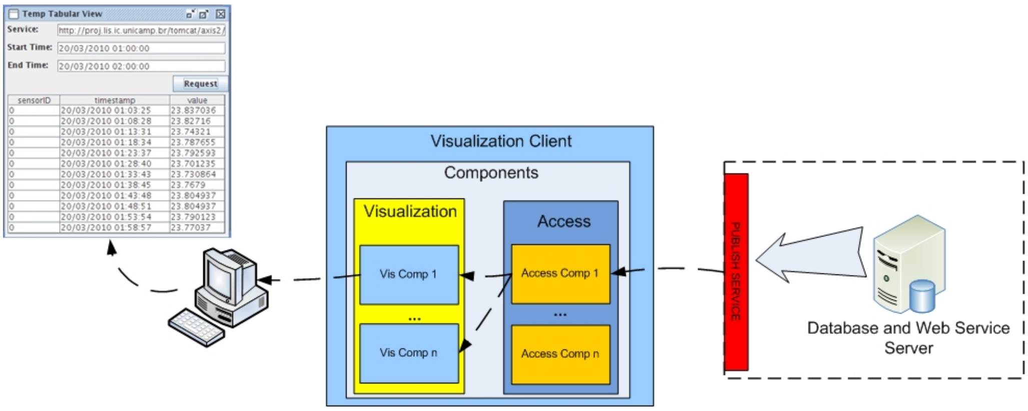

Figure 3. Components to request published data and visualize it.

Our solution combines aspects from NOSA and Pastorello's work (see section 2). From the latter, it adopts the philosophy of components to encapsulate functionality and increase modularity. From NOSA, it uses aspects of publication using Web Services, thereby increasing interoperability. Moreover, we treat the heterogeneity problem at the storage level, standardizing the format to store sensor data. Thus, the role of components is to provide distinct visualization formats, including simultaneous views of multiple sensors.

Our solution also provides flexibility for multiple kinds of queries – eg, involving aggregation, interpolations or transformations. Each such request can be implemented by adding new methods at the Publication service level, mapped into PostgreSQL queries. Then, visualizations components can invoke methods. Alternatively, we can also envisage a chain of components that implement such functions on top of basic invocations of the Publication service (i.e., instead of changing the service we add components). In such a case, a request for an aggregation over a period of time for x temperature measurements is translated into an execution of a sequence of components – the first will request from the Publication data for the sensors and period desired and the second will compute the aggregation.

4. Case Study: a hands-on experiment

Our case study concerns managing data from a WSN deployed at the Faculty of Agriculture Engineering (FEAGRI) at UNICAMP. Sensor data are collected at an access point installed at FEAGRI, to be processed at the Laboratory of Information Systems (LIS), at the Institute of Computing. For this experiment, we were concerned with basically three issues – sensors heterogeneity, data publication and processing. To create a heterogeneous test case, we collect data from sensors sensing air humidity, light and temperature. Even with a small amount of sensors, we had to deal with 3 types of measurement, each with different frequency of data acquisition and units.

All the sensors send data to a local base station which is connected to a computer that has a web service client (the Data Load service). This WS client, developed by us, sends the measured sensor data to a server located at LIS. This WS server (Receive Service) at LIS then stores the data in an appropriate way, and finally publishes data through another WS (Publication Service).

For temperature measurements, our extension to FLAVOR consisted in creating two components: TempSensorAccess and the TemperatureSensorTabularView, respectively the access and the visualization components for accessing the temperature sensor data available. Figure 4 shows an example of visualization of temperature data, using a data table format, using these components. An alternative means of visualizing the sensor data appears in Figure 5. This was developed using a distinct software, but using the same underlying data stored in PostgreSQL. Such flexibility in handling data is only possible because of our architectural choices.

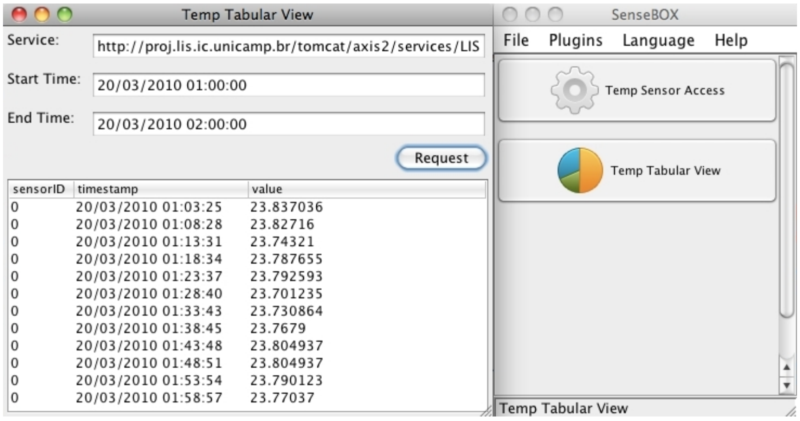

Figure 4. Client visualizing Temperature sensor data in a data table.

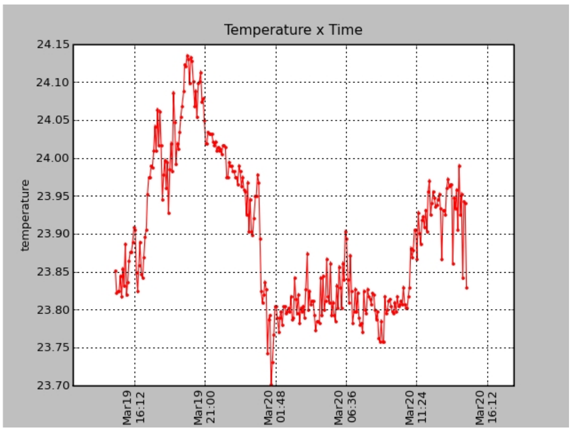

Figure 5. Temporal Series showing another view of some of the data presented in Figure 4.

So far we are receiving data from 5 sensors, which will be extended to 11 sensors by the end of March 2010. Another network with 32 sensors will be deployed by mid-April. There were several difficulties in deploying the first sensors, ranging from engineering problems to defining a storage format for data. For instance, the setting up of the communications infrastructure and the calibration of sensors took more than one year.

Web Services was one of the many solutions discussed for communication between layers. Again, service specification and implementation was time consuming. However, once the services were running and the network was deployed, the extension of access and visualization alternatives is proving to be relatively straightforward, because of the architectural solution adopted.

5. Conclusions and ongoing work

This paper presented our proposal to process and visualize sensor network data. Our approach is based on combining web services (to provide interoperability among sensor networks, data servers and user applications) and components (to provide flexibility in data preprocessing and visualization).

There are many directions for continuing this work, that range from solving issues such as detecting faulty sensors to providing users with a wide range of visualization and filtering options.

6. Acknowledgements

This work was partially financed by the Microsoft Research Fapesp Virtual Institute (eFarms project), CNPq (BioCORE project) and CAPES, as weel as a project grant from CNPq.

References

Aberer, K., Hauswirthand, M., and Salehi, A. (2007). Infrastructure for data processing in large-scale interconnected sensor networks. In *Proceedings of the Mobile Data Management (MDM 2007)*, Mannheim, Germany. IEEE Computer Society.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38:393–422.

Chu, X., Kobialka, T., Durnota, B., and Buyya, R. (2006). Open sensor web architecture: Core services. In *Proceedings of the 4th International Conference on Intel ligent Sensing and Information Processing (ICISIP 2006)*, pages 98–103, Piscataway, New Jersey, USA. IEEE Press.

Hey, T., Tansley, S., and Tolley, K., editors (2009). *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Corporation.

Huang, C.-F., Tseng, Y.-C., and Wu, H.-L. (2007). Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network. *ACM Trans. Sen. Netw.*, 3(1):5.

Koga, I. K., Sampaio, L., and Monteiro, J. A. S. (2007). FLAVOR: A dynamic and open framework for the development of network measurement access and visualization tools. In *XXV Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*, pages 665–678, Belém, PA.

Krasner, G. E. and Pope, S. T. (1988). A cookbook for using the model-view controller user interface paradigm in smalltalk-80. *J. Object Oriented Program.*, 1(3):26–49.

Medeiros, C. (2008). Grand Research Challenges in Computer Science in Brazil. *IEEE Computer Society*, 41(6):59 –65.

Nakamura, E. F., Loureiro, A. A. F., and Frery, A. C. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput. Surv.*, 39(3):9.

OGC (2010). Sensor Web Enablement. http://www.opengeospatial.org/projects/groups/sensorweb.

Pantazis, N. A., Nikolidakis, S. A., and Vergados, D. D. (2009). Energy-efficient routing protocols in wireless sensor networks for health communication systems. In *PETRA '09: Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments*, pages 1–8, New York, NY, USA. ACM.

Pastorello Jr, G. Z., Medeiros, C. B., and Santanchè, A. (2007). Providing homogeneous access for sensor data management. Technical Report IC-07-012, Institute of Computing, State University of Campinas.

Santanchè, A. and Medeiros, C. B. (2007). A component model and an infrastructure for the fluid web. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):324–341.

Zhao, Q. and Gurusamy, M. (2008). Lifetime maximization for connected target coverage in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 16(6):1378–1391.