Enhanced Service-Oriented Open Sensor Web Architecture with Application Server Based Mashup

Muhammad Sohail Khan and DoHyeun Kim

College of Computer Engineering, Jeju National University, 102 Jejudaehakno, Jeju-si 690-756, Republic of Korea

Correspondence should be addressed to DoHyeun Kim; kimdh@jejunu.ac.kr

Received 4 March 2014; Accepted 26 May 2014; Published 24 June 2014

Copyright © 2014 M. S. Khan and D. Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Connecting the sensing devices present in the physical world to detect and measure various physical phenomenon such as temperature, humidity, and pollution into a network and presenting them as web resources to the end users have become the goal of a variety of research activities. As the physical network of these devices inherently possesses a heterogeneous nature thus most of the sensor web studies are based on providing domain specific solutions. Service-oriented architecture (SOA) has proven successes to resolve the scalability and evolvability of the web. The present solutions for the SOA based sensor web allows the users to access the sensor data as services but to integrate these services the client must put extra effort to implement the logic. We present an improved SOA based sensor web architecture which offers an easy approach to integrate sensor providers’ services with information provider services and enable the users to access it as a single, integrated, and searchable service. This approach offers the functionality present in the existing solutions as well as an integrated data access or service mashup. We discuss in detail an instance architecture for indoor environment monitoring, based on the proposition of this study.

1. Introduction

Sensors are devices which detect, sense/measure, a physical phenomenon, that is, light, temperature, motion, humidity, and so forth [1]. These devices also have the capability to measure a broad range of these phenomena with a considerable amount of accuracy [2]. The current advancements in the microelectronics and nanotechnology sectors are already proving to be correct, the predictions made by Perera et al. [3] that all personal objects such as phones, wrist watches, lighters, handheld devices, and almost all households will have embedded sensors and will be used to retrieve information regarding various factors of their surroundings. To use these sensing devices for the production of valuable information about our surroundings and the phenomenon occurring around the world, these devices are being combined into networks. Basically sensor networks are composed of smaller but long running sensing nodes with some amount of computing capabilities. The sensing nodes work together to collect information such as temperature, humidity, images, and light intensity. The accurate and reliable data collection of these sensor networks has enabled the research community to develop sensor networks based applications for environmental factor monitoring, context prediction, and early warning systems for floods, forest fires, earthquakes, and so forth [4, 5].

There are various technologies in existence such as Zigbee, WiFi, and Bluetooth through which sensors can interact with each other to form a sensor network and to communicate the data. The hardware for different sensor types and the communication technologies employed by various sensing devices add a heterogeneous feature to the sensor networks. In such a heterogeneous composition of a sensor network, the collection of data from different sensing nodes becomes a challenging task. Various infrastructures are being devised and proposed in order to manage these networks efficiently, integrate the data from these networks, and make the data easily available to the end users. The Open Geospatial Consortium (OGC) has presented standards with focus on sensors and networks of sensors called Sensor Web Enablement (SWE). Today, the term “sensor web” has mostly been influenced by the developments of SWE initiative [6].
and is defined as an infrastructure which allows the users to share sensor resources in a well-defined way by hiding the underlying details of network communications and sensing hardware. SWE defines sensor web as "web accessible sensor networks and archived sensor data that can be discovered and accessed using standard protocols and application programming interfaces” [7].

SWE provides the service specifications for the enablement of sensor web. Instance architecture of the OGC SWE is shown in Figure 1 presenting the layers and services specified by SWE. The detailed specifications of each service can be found in [7]. This paper presents a similar but enhanced service-oriented architectural framework based on two concepts: (a) service provider based sensor data provision as in the existing solutions and (b) application server based binding of the service providers (services mashup) which enables the users to access data from multiple service providers as a single integrated service. The first concept enables the archiving of sensor data from the sensor gateway which provides abstraction from the underlying heterogeneity of the sensor networks. It also makes possible the provisioning of sensor data to the users based on the exposed services. The second concept is the proposed enhancement of the existing solutions providing the integration layer for desperate sensor networks and other information providers (i.e., spatial, temporal) based on the binding of data from the respective service providers. This application layer data binding through the provider services offers an easy and efficient integration of sensor data which when exposed as a single service, enables the clients to use and visualize the data without any extra implementation at the client application.

The rest of the paper is organized as follows. Section 2 provides a brief overview of the related work in terms of the different approaches and frameworks implemented in order to connect sensors and applications. Section 3 describes the service provider based sensor web and the application server based sensor web conceptual frameworks. It also contains the detailed description of the data collection process and data provisioning to the clients for both frameworks. Section 4 provides a detailed description of the proposed sensor web architecture. An instance implementation of the proposed architecture for location based indoor environment monitoring is discussed in Section 5. This section contains a detailed illustration of each module in the instance architecture. The paper is concluded in Section 6 with some future works included in the same section.

2. Related Work

The current research in the field of sensor web is intended towards one major goal and that goal is to make the sensing
resources available to the internet applications. To meet this goal, several technologies and infrastructures have been developed to aid in the management of the heterogeneity of the sensing resources and their networks thus enabling the application layer to use these resources in an easy and efficient way. This section presents a brief overview of the related work in terms of the infrastructures and approaches being adopted by the research community so far to achieve the ultimate goal of sensor web. Bröring et al. [6] present a comprehensive review of the current sensor web middleware frameworks and infrastructures. Sensor web can be considered as a bridge between the physical sensing devices and the application using the data produced by these devices. According to Bröring, this bridging area between the actual sensing devices and the applications to use their data is divided into three architectural layers, namely, sensor layer, sensor web layer, and application layer [6]. Sensor layer provides the abstraction from the underlying heterogeneous technologies employed by the sensing devices and the protocols to communicate with in the network of the sensing devices. Sensor web layer normally implements the strategy through which the bridging between physical sensing devices and application’s data requirement is achieved. Finally, the application layer provides the interface for the users (humans or other systems) to search, utilize, and interact with the resources (in the form of services) made available by the sensor web layer.

Sensor web concept starts with networking of sensing devices and the management of these networks. Several aspects of the sensor networks, such as communication within the networks and optimization (energy, coverage area), are the lowest levels of studies performed related to sensor web. A few of such studies which are specific to the wireless sensor networks (WSN) include the intranetwork communication optimization as described in [8], optimization of coverage range for the sensor networks [3, 4], and intranetwork sensor localization as reported by [9]. Similarly, studies of routing protocols in WSNs have been reported by [10]. A detailed survey of the different approaches for the development of WSN management infrastructures has been reported by Wang et al. [11]. As these solutions are focused on the management issues of sensor networks at the lower level thus they can only be used as basis (abstraction layer) while envisioning the overall architecture for the sensor web infrastructure from device to client.

The second major area of research regarding sensor web framework architecture is focused on solutions to make sensing resources available all the way up to the clients at the application layer. While such studies focus on the strategies and approaches to make the sensor produced data accessible to the clients in an efficient and easy way, they abstract out the lower level issues of sensor network management. The most well-known work in this group of sensor web framework is the first implementation of Sensor Web Enablement (SWE) service specifications by the Open Geospatial Consortium (OGC). This implementation is named as 52° North sensor web framework [12]. 52° North framework uses an intermediate layer named as the sensor bus in order to provide the integration of sensing resources with the implementation of SWE services [13]. The SWE service specifications have also been used by several other projects. For example the GeoSWIFT 2.0 [14] implemented the SWE services and provided extended scalability based on the peer-peer spatial query framework; a combined implementation of SWE services with the Web 2.0 technology was envisioned in the NASA’s sensor web 2.0 [15], for the creation of mash-up applications to integrate the data from multiple sources. Representational state transfer (REST) services were used for the integration of data but according to [6], the provision of REST access to the sensing resources is unclear.

Global sensor network (GSN) [16], SOCRADES [17], and Hourglass [18] are some of the famous studies for the development of sensor web frameworks which does not follow any standardized approach in their implementation. For example, GSN focuses on the abstraction of sensors (virtual sensors) with the help of XML descriptors and used simple distributed SQL queries for data access. The same approach is followed by Hourglass providing the framework for connecting applications and sensing devices. SOCRADES provides a service-oriented implementation by exposing sensor functionality as service and integrating these services to the sensor web architecture by implementing sensor gateways to abstract out the underlying communication technologies.

Another class of emerging systems is the centralized sensor web enabling users to contribute sensor data to the centralized system. The data formats which can be shared by the users are dependent on the system and it may be sensing data in the form of numbers (temp, humidity, etc.) or audio, video data. The uploaded data can then be queried and utilized by the end users. Sense Web [19], SensorBase [20], and Sensorpedia [21] are few instances of such systems. These centralized sensor web systems provide APIs for the registration of sensors owned by the users, uploading data and querying the uploaded sensor data. There are various issues with such infrastructures such as the hosting of sensor data at the central system instead of separate service modules
making the system unsuitable for scenarios demanding full control over deployment and data privacy.

The most recent focus of research community is towards the integration of networks into a common network of things called the Internet of Things [22]. This vision bears the concept of bringing into a communication network, not sensing devices but also other identifiable things such as RFID tagged objects as well as information resources. We have discussed various classes of infrastructures and architectures for the implementation of sensor web and found that most of them address a specific domain of usage scenarios. Service orientation has already opened the ventures for investigating the integration of these domain specific infrastructures. The potentials of a single integrated architecture of sensor networks and sensor web infrastructures have been explored in this paper.

3. Conceptual Framework

In this section we present the conceptual framework for the enhanced service-oriented architecture for sensor web systems. The architecture provides the description of service orientation such as minimal device and device network dependence, service publishing and discovery functions, and easy device registration and device control features. The architecture can be visualized from two different points of view based on the level at which the services to the clients are being provided, that is, service provider based design and application server based design. The following subsections describe each point of view in detail.

3.1. Sensor Web Architecture Based on Service Provider.

Figure 2 presents a conceptual framework of the service provider based sensor web architecture. Sensors represent the network of sensors and other RFID tagged objects. Gateway or the middleware is the layer which is responsible for the communication and interaction of upper layers of the system with the physical sensors. Gateway/middleware is the implementation of drivers and communication channels which are used to interact with the sensors in order to collect sensed data from the sensors and state data specifying their connectivity to the system.

As mentioned, gateways only provide connection and communication point to the sensing devices in the physical world and the actual data about the registration, state of these devices, and the data produced by these sensing devices resides at an upper layer called the service provider. Service provider maintains an information repository for this purpose. Therefore, whenever the information or data is required by the system or the clients connected to it through the exposed services, the recent data from the repository is delivered to the requesting entities.

The services exposed by the service provider are atomic in nature and are web service description language (WSDL)
 enabled. Hence, these services can easily be published to a service registry. At the service registry, the clients can search for the required services and consume them according to their needs. Figure 3 illustrates the process of sensor web data collection and the data provisioning process through a sequence diagram.

3.2. Sensor Web Architecture with Application Server Based Service Mashup. The conceptual framework for the sensor web architecture based on service providers shows that the client can utilize the services exposed by a variety of sensor service providers but to combine or integrate the data acquired from different service providers, the client must implement its own logic. The implementation of logic at the client application for the integration of data from service providers is not a feasible solution especially when we are talking about users with no programming experience. Although, it is important to provide sensing resources as atomic services which can be combined at the application level according to the will (requirements) of the developers there must be some way to provide integrated services which may act as a complete solution and the clients just need to utilize it without having to implement any logic.

Figure 4 presents the conceptual framework for the application server based enhanced sensor web architecture. It offers the integration of data from different service providers based on the information binding and presenting it to the users as a single mashed-up and searchable service. The schema for binding the data is defined by each application server according to the features of the solution it provides and the data is exposed as a single service to the client. This way the client does not need to implement any integration or binding logic and simplifies the implementation of client application. New providers and application servers can be plugged into the system without affecting the existing system. The client is able to consume any service offered by the service providers or the application server thus enhancing the potentials of the existing sensor web architecture.

The rest of the architecture is similar. The gateway/middleware provides connectivity and communication functionality to the underlying sensor network providing abstraction from the heterogeneity of the hardware and technology used.

Service providers, through the exposed services, collect the sensor data from the sensor network via the middleware. Service providers store the data and state of each connected sensing device and expose a provider service which can be consumed by a client directly or used by an application server.

Application server is defined on top of the service providers. According to the specifications and requirements of the specific application server, the provider services of the underlying service providers are consumed to define the schema for integration of the data. Once a schema is defined, a single integrated service is defined and exposed so that client can consume it to use the integrated data from multiple service providers. An example scenario of application server based integration of sensor service provider and GIS service provider has been illustrated in Section 5.

4. Proposed Architecture

Figure 5 shows the enhanced open sensor web architecture. As in the standard SWE sensor web architecture, there are three basic layers, namely, physical layer, middleware, and service layer. Physical layer consists of the actual hardware and networks of the sensing devices. The middleware layer acts as an abstraction layer and the gateways at this layer implement all the necessary technologies and drivers to communicate with the underlying network of sensing devices. At this layer, the main concerns are abstracting the heterogeneity of the underlying networks and the easy deployment of new sensors in the networks. The proposed architecture envisions dedicated gateways for different types of sensing devices. Thus each gateway is responsible for communicating with a homogeneous sensing network which is easy to manage and adding or removing new sensors to the network.

The service providers expose services in order to fill the gap between gateways and the application server on top of the service layer. The services exposed by the service provider are sensing service, content service, and provider service. The sensing service offers the functionality of data collection from the underlying gateways and storing the sensor data in the repository at the provider level. The content service offers the functionality to manage the information repository regarding the connected gateway and network of sensors to a provider. The provider service offers the functionality of querying the sensor data and information from service provider repository.

The application layer consists of application servers which consume the services exposed by the underlying service
providers in order to bind and integrate the data according to the requirements of the application. The integrated data is finally exposed by the application server as a mashed-up service and published to the service registry. Client can search these services, published by one or more application servers, and may choose a service according to their required data using a simple client application. The simplicity of the client application comes from the fact that it only needs to consume a service in order to get the combined data of one or more service providers.

The scalability of the system is insured by the fact that any number of service providers at the service layer can be added with their underlying infrastructure of middleware and sensor networks. The services from these providers can be made available easily to the clients by publishing to the service registry and these services can also be used at the application layer by the existing application servers or new application servers added for specific solutions. Along with the distribution of the architecture, the application layer also provides for centralized processing of the integrated data from different providers.

5. Instance Architecture

Here we present the detailed instance implementation of the improved service-oriented open sensor web architecture. This instance implementation is based on the scenario of an integrated sensor web for monitoring the indoor environment of a building.

The scenario for the instance architecture considers indoor environment monitoring based on indoor locations. Thus the appropriate solution for the scenario would be to integrate the data produced by the sensing devices inside the building with the location data from a GIS service provider.

The major modules of the design are shown in Figure 6 while the process of interactions among these modules has been presented in Figure 7. Each module has been described briefly in the following sub sections.

5.1. Sensor Middleware. Sensor Middleware is the component which directly connects to the sensor emulator or the actual sensor network. It provides an abstraction layer to overcome the heterogeneity of the underlying sensor devices or the network of such sensing devices. It has the drivers (software) to establish connection with the physical sensing devices and get the sensing data produced by these devices. The sensed data collected from the sensors at a predefined interval of time is forwarded by the middleware to the sensor web provider.

5.2. Service Provider. Sensor web provider acts as the sink for sensed data in its raw form produced by the sensor networks connected to it through one or more middleware. This module also acts as a service provider for clients of the same data. It exposes services such as sensor web content service for the management of information regarding the registered sensor nodes and the middleware configuration, Sensing service for the collection of raw sensed data from the middleware associated with the service provider and a data provision service exposed for the clients or any application which needs the sensed data. It has often been argued that sensing nodes are resource limited devices which often turn off themselves to conserve energy [23]. In such a scenario when the sensing node is off for any reason, the behavior of the system becomes important. To tackle this issue, in the proposed architecture, the service provider maintains a repository of sensing data along with the information for each sensing node. So the clients get the sensing data from...
5.3. **GIS Provider.** We are presenting the architecture of an integrated sensor web system based on a scenario where the sensing data and the location information services are being combined for the clients. Thus, geographic information system (GIS) must be a part of the system if the sensor information is to be integrated with the location information. To meet this requirement, we propose an integrated GIS provider which manages the record of the maps and the map contents such as building info, floor info, and room info. The GIS provider exposes services such as content management service and provider service. The content management service can be utilized for the management (create, delete, and update) of map contents and the related information. The provider service may be used directly by the clients to get information related to only the buildings utilizing the system or it may be used by an application server to integrate the GIS information with the sensor information from the respective providers.

5.4. **Service Registry.** Service registry module basically provides service based interface for publishing services exposed by service providers such as application server provider, sensor service providers, and GIS service providers. For this purpose, service registry maintains a service information repository where the published services can be stored and maintained (enabled/disabled) from a centralized location. Similarly, it also exposes a search service interface for the clients to search services published by service providers.
The search service takes the user query (based on keywords) to provide the information about services that are already published by service providers and displays it as a list. The client is provided with basic information about the service through which it can utilize the service and send or receive the intended information from the specific service provider. The registry module also provides service management functionality where services already published can be made hidden and deleted from the record or hidden services can be published again. Hiding a service means that the information about that service exists in the record but these services are marked as inactive and the service information will not be returned in response of a search query.

5.5. Application Server. Application server provides the interface for combining the data offered by service providers. It also offers services to the clients to deliver them with the combined and processed data from service providers. Figure 8 shows the main interface for the indoor environment monitoring application server. The scenario upon which the sensor web architecture presented in this paper is based requires the binding of data from both sensor service provider and GIS service provider. The binding of data is performed by getting the sensor information and sensed data from the sensor service provider while the location information is obtained from the GIS service provider. Based on the information from the two service providers, a combined visualization service is exposed to the client. The client utilizes this service to view the sensing devices on a map of the selected building. The user may select the sensing devices associated with a specific indoor location. Based on the client selection the appropriate provider service is used to fetch the data produced by the sensing device and delivered to the client after or without processing. Application server can also perform the functionality of centralized decision and control module where an intelligent algorithm may run to evaluate the current state of the underlying sensor networks and autonomous control commands can be issued to the devices in the network. Table 1 summarizes the major functions for the implementation of the application server in this scenario.

5.6. Client. AppServer client is a web based application to enable the users to access the sensor web system using
Table 1: Application server API for indoor environment monitoring.

| Class                | Function                      | Explanation                                      |
|----------------------|-------------------------------|--------------------------------------------------|
| frmMainPage          | TimeNowTick                   | Update the system time                           |
|                      | ProTimeTick                   | Update work time of provider service             |
|                      | ConTimeTick                   | Update work time of content service              |
|                      | CtrlTimeTick                  | Update work time of control service              |
|                      | GetSensorNum                  | Get sensor object number from the database        |
|                      | ControlRooms                  | Get room list from DB                            |
| AppProviderService   | GetMapServiceUri              | Get map service information from the database     |
|                      | GetObjectList                 | Get object information from the database          |
|                      | GetRoomEnvironment            | Get comfort index of some room                    |
| AppContentService    | GetMapServiceUri              | Get map service information from the database     |
|                      | IsExistMapService             | Check the map service this is be binded yes or no |
|                      | DeleteMapServiceBinding       | Delete map service binding information from the database |
|                      | SaveMapServiceUri             | Save map service information into the database    |
|                      | IsExistObject                 | Check whether object information existed in the database |
|                      | AddObject                     | Save object detailed information into the database |
|                      | SetObject                     | Change object detailed information in the database |
|                      | DelObject                     | Delete object detailed information from the database |
|                      | GetObjectCode                 | Get object code by object ID and service provider's Uri address from the database |
|                      | GetObjectList                 | Get object detailed information in a part of floor map from the database |
|                      | GetObjectName                 | Get object information number from the database   |

a web browser. The AppServer client searches the available AppServer provider services at the service registry and connects to one of them. It acquires the map service URI from the AppServer and gets the respective map information from GIS provider using that URI. It provides an interface where the users can view GIS contents and the bound sensor nodes. It also acquires the data of a particular sensor and also displays the comfort index. Figure 9 shows the visualization of the integrated indoor environment monitoring service through a web client. The map shows the associated sensing devices and upon selection area "A" displays the sensing information and data for the selected node.

6. Conclusions

This paper proposes improved service-oriented sensor web architecture. The proposed architecture is capable of data provision to the end users of the system at two layers. The service provider based data provision through exposed services enables the users to access raw sensor data from the service providers in near real-time and the data is archived and can be accessed collectively according to the queries of the users. The application server provides the integration of data from different sensor service providers and information service providers. It also implements logic for data processing and decision support on the integrated data and exposes it to the users as a single service. This architecture offers easy implementation of client applications which only needs to implement the services offered by various application servers. The paper also presents an instance architecture based on the scenario of indoor environment monitoring system with the integration of data from two service providers, that is, sensor service provider and GIS service provider.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Industrial Strategic Technology Development Program (no. 10043907, Development of high performance IoT device and Open Platform with Intelligent Software) and funded by the Ministry of Science, ICT & Future Planning (MSIF, Korea). This work was supported by MOTIE-KETEP, Republic of Korea (Project no. 201185100010040, Development of proactive energy management technologies for saving the energy cost of residential complex using cooperative sensing devices).
References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, 2002.

[2] libelium Comunicaciones Distribuidas, “libelium,” 2006, http://www.libelium.com/.

[3] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Sensing as a service model for smart cities supported by internet of things,” Transactions on Emerging Telecommunications Technologies, vol. 25, no. 1, pp. 81–93, 2014.

[4] H. Conover, G. Berthiau, M. Botts et al., “Using sensor web protocols for environmental data acquisition and management,” Ecological Informatics, vol. 5, no. 1, pp. 32–41, 2010.

[5] J. Hayes, E. O’Conor, J. Cleary et al., “Views from the coalface: chemo-sensors, sensor networks and the semantic sensor web,” in Proceedings of the International Workshop on the Semantic Sensor Web (SemSensWeb ’09), 2009.

[6] A. Bröring, J. Echterhoff, S. Jirka et al., “New generation sensor web enablement,” Sensors, vol. 11, no. 3, pp. 2652–2699, 2011.

[7] M. Botts, G. Percivall, C. Reed, and J. Davidson, “OGC® sensor web enablement: overview and high level architecture,” in GeoSensor Networks, pp. 175–190, Springer, Berlin, Germany, 2008.

[8] S. Lin, V. Kalogeraki, D. Gunopulos, and S. Lonardi, “Efficient information compression in sensor networks,” International Journal of Sensor Networks, vol. 1, no. 3, pp. 229–240, 2006.

[9] K.-Y. Cheng, K.-S. Lui, and V. Tam, “HyBloc: localization in sensor networks with adverse anchor placement,” Sensors, vol. 9, no. 1, pp. 253–280, 2009.

[10] K. Akkaya and M. Younus, “A survey on routing protocols for wireless sensor networks,” Ad Hoc Networks, vol. 3, no. 3, pp. 325–349, 2005.

[11] M.-M. Wang, J.-N. Cao, J. Li, and S. K. Dasi, “Middleware for wireless sensor networks: a survey,” Journal of Computer Science and Technology, vol. 23, no. 3, pp. 305–326, 2008.

[12] 2013, http://52north.org/swe.

[13] A. Broering, T. Foerster, S. Jirka, and C. Priess, “Sensor bus: an intermediary layer for linking geosensors and the sensor web,” in Proceedings of the Ist International Conference and Exhibition on Computing for Geospatial Research and Application (COM.Geo ’09), June 2010.

[14] S. H. L. Liang, “A new peer-to-peer-based interoperable spatial sensor web architecture,” Power, vol. 3, pp. 4–1, 2008.

[15] D. Mandl, P. Cappellaere, S. Frey et al., “Sensor web 2.0: connecting earth’s sensors via the internet,” in Proceedings of the NASA Earth Science Technology Conference, 2008.

[16] K. Aberer, M. Hauswirth, and A. Salehi, “A middleware for fast and flexible sensor network deployment,” in Proceedings of the 32nd International Conference on Very Large Data Bases, pp. 1199–1202, VLDB Endowment, 2006.

[17] L. M. de Souza Sá, P. Spiess, D. Guinard, M. Köhler, S. Karnouskos, and D. Savio, “Socrates: a web service based shop floor integration infrastructure,” in The Internet of Things, pp. 50–67, Springer, Berlin, Germany, 2008.

[18] J. Shneidman, P. Pietzuch, J. Ledlie, M. Roussopoulos, M. Seltzer, and M. Welsh, “Hourglass: an infrastructure for connecting sensor networks and applications,” Harvard Technical Report TR-21-04, 2004.

[19] A. Kansal, S. Nath, J. Liu, and F. Zhao, “SenseWeb: an infrastructure for shared sensing,” IEEE Multimedia, vol. 14, no. 4, pp. 8–13, 2007.

[20] K. Chang, N. Yau, M. Hansen, and D. Estrin, “A centralized repository to store sensor network data,” in Proceedings of the International Conference on Distributed Computing in Sensor Systems, June 2006.

[21] B. L. Gorman, D. R. Resseguie, and C. Tomkins-Tinch, “Sensorpedia: information sharing across incompatible sensor systems,” in Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS ’09), pp. 448–454, May 2009.

[22] L. Atzori, A. Iera, and G. Morabito, “The internet of things: a survey,” Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[23] G. Anastasi, M. Conti, M. di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: a survey,” Ad Hoc Networks, vol. 7, no. 3, pp. 537–568, 2009.