J. Sens. Actuator Netw. **2013**, *2*, 98-108; doi:10.3390/jsan2010098

OPEN ACCESS

Journal of Sensor and Actuator Networks

ISSN 2224-2708

www.mdpi.com/journal/jsan

Review

A Survey of Sensor Web Services for the Smart Grid

Omar Asad, Melike Erol-Kantarci and Hussein T. Mouftah *

School of Electrical Engineering and Computer Science, University of Ottawa, 800 Avenue King Edward, Ottawa, ON K1N 6N5, Canada; E-Mails: omar.asad@uottawa.ca (O.A.) merolka2@uottawa.ca (M.E.-K.)

* Author to whom correspondence should be addressed; E-Mail: mouftah@uottawa.ca; Tel.: +1-613-562-5800 (ext. 2173); Fax: +1-613-562-5664.

Received: 12 December 2012; in revised form: 26 January 2013 / Accepted: 9 February 2013 / Published: 6 March 2013

Abstract: The broad use of Wireless Sensor Networks (WSN) in various fields have resulted in growing demand for advanced data collection and querying mechanisms embedded in the sensor node. Sensor Web Services (SWS) have recently emerged as a promising tool to enable external machines to have access to the information collected by public sensor webs. Machine-to-machine interactions or wireless sensor and actor networks can take advantage of this platform-independent technology to develop diverse smart grid applications. In this survey, we first briefly present the state of the art in SWS technology by describing the techniques for customizing web services to fit the sensor node capabilities such as customizing the WSDL file, compressing XML documents and redesigning TCP protocol. Then, we survey the studies that have utilized the SWS technology in smart grid applications. These studies have shown that SWS provide energy management capabilities to the consumers and the utilities, and they are well suited for smart grid integrated smart home solutions.

Keywords: sensor web services; smart grid; smart home; wireless sensor networks

---

1. Introduction

Wireless Sensor Network (WSN) is a network of low-cost sensor nodes which have the ability of collecting data from their surroundings and transmitting those data to their peers or a sink node via

multi-hop wireless links. Publicly available WSNs also have the capability of forming the sensor web which can be considered as an extension of the future internet towards smart devices, *i.e.*, Internet of Things (IoT). By the use of sensor web services, physical information from smart homes, Plug-in Hybrid Electric Vehicles (PHEV) will be available online to the owners [1–3]. Besides data collection and querying capabilities, sensor web services can provide remote management capabilities, which enriches the ability of a wireless sensor and actor network to respond to events in the smart home and the smart grid and in a broader concept, a smart planet [4]. references should be in numerical order

In the past few years, many works in the literature have targeted the area of the communication paradigm within the smart grid [5–7]. However, the focus of these works is mainly on the network layer, which can provide a channel of communication among different smart grid components. Nevertheless, this survey is addressing a powerful, yet an elegant tool of WSNs' data communication within the domain of the smart grid, and to the best of our knowledge, this is the first survey paper on sensor web services in the context of smart grid.

A web service can be defined as a remote method that resides on an electronic device that is accessible through the web. Therefore, Sensor Web Services (SWS) describes the procedure of implementing a web service in a certain wireless sensor node that is able to provide a tool for data exchange with the node itself and external web-machine.

The advantages of implementing web services on sensor nodes are as follows. SWS increase the interoperability of the WSN with the applications by elevating the need for the developers to know the details of the platform. SWS improves the ability to develop higher level applications, as the Web Service Description Language (WSDL) file contains information about the web service methods which can be used by a high level development tool like NetBeans and Net. This property provides easy integration with web enterprise applications, especially considering that web services are very common in Internet applications. Finally, when web services are implemented on the sensor nodes, the need for a gateway between the external machine and the sensor nodes is elevated.

Besides the above mentioned advantages, using web services on resource-constrained sensor nodes is challenging due to several reasons. First, the payload of the web service messages consists of XML data, TCP/IP, and HTTP headers, which can easily exceed the memory limitations of a sensor node as well as the sensor network transmission capabilities. For instance, the Pacemate sensor network hardware platform [8] only has a 32 KB RAM while a single XML message can be as much as 500 KB. Furthermore, the size of the code and the compiler has to fit in the sensor memory. Hence, previous research have focused on customizing the web services and its components to fit into the limited resources of a sensor node. Recently, several studies have employed sensor web services to smart grid applications to enhance energy control and management systems. In this paper, we provide a brief summary of the studies that focus on extending the conventional web services for the use of sensor web. We first summarize the techniques that rely on customizing the web service components and then, we present a detailed review of the techniques that focus on modifying the transport protocol. Furthermore, we survey the studies that adopt SWS technology in smart grid applications. Sensor web is anticipated to be widely deployed in the homes, vehicles, buildings and the power grid: hence by the help of web services, sensor network can serve as a tool to coordinate and control power demand from appliances, PHEVs, Air Conditioning (AC) appliances, *etc*.

The rest of the paper is organized as follows. We give a brief summary of the sensor web services in Section 2. In Section 4, we present the employment of sensor web services in smart grid applications. Section 6 concludes our paper.

2. Sensor Web Services

The idea of the sensor web, where the sensing data could be accessed through World Wide Web (WWW), has been initially proposed by NASA's Jet Propulsion Laboratory(JPL) in order to explore planets in unknown environments [1]. Later, sensor web services has been initially tested in a green house to measure the soil temperature, humidity, and light level in [9]. An architecture for a worldwide sensor web has been formed in [2], to provide connectivity to a heterogeneous sensor network trough the web. Following these studies, in [10], the authors have formed the sensor web technology as a Service Oriented Architecture (SOA), where each sensor node carried out a service published by itself through the web. Furthermore, in [11], Woo *et al*. have presented a demo of sensor web services where the SensorNet server publishes a WSDL file that exposes the system and application services on the sensor nodes. The purpose of this study was to demonstrate that different languages and programming models could be used to develop applications on the sensor nodes. Furthermore, sensor web services have been employed to an emergency health application in [12]. The proposed application enables the local personnel with smart phones to access the sensor data via web services in order to provide the necessary pre-hospital patient care.

Sensor web services provide opportunities for various applications, some of which has been briefly summarized above. In the following section, we discuss the solutions that address the challenges related with implementing web services on a sensor node. Sensor web services are different with the conventional web services designed for powerful machines, because sensors have limited resources both in terms of memory and energy. For this reason, research on sensor web services has focused on customizing the web service components in order to tackle the limited-resource constraint of the sensor nodes. In the following section, we discuss web service customization techniques.

2.1. Customizing XML

Extensible Markup Language (XML) is the data structure used to encapsulate the method names and arguments on the web services. The payload of the XML file is relatively large. Therefore, the use of XML data structure on sensor nodes requires modifying the XML file to fit the node constrains.

In IrisNet [2], the authors have represented the sensor node in an XML format depending on its regional or political locations such that the WSN is divided into several parts and each part is divided into smaller subparts. When the XML file is structured to represent the WSN, the large area possesses the main tag id and each subarea has a sub tag that enables to reach the individual sensor node.

Sensor Model Language Encoding Standard (SensorML) is an information model introduced by [13] that was built using XML data structure to give a description of a sensor node hierarchy in XML language. The model provides a description for any process carried out by the sensor node, like the measurement process and instructions for accessing high level information from the surrounded

phenomenon. The processes described by the XML can be discovered and executed. Therefore, all processes define their inputs, methods, and outputs.

Augeri *et al.* [14] have given an analysis for the XML compressor tools and identified the key factors when choosing the compressor. XML tree structure compression has been discussed in [15]. The idea of the corresponding study has been to compress the tree structure part of XML document, not the data content. The authors have used an algorithm to derive the grammar of the tree structure depending on the repetition of the tree pattern.

In [16], the authors have optimized XML by replacing some of the method names and argument names defined by the sensor node with very compact names.

In [8], the authors have used the Business Process Execution Language (BPEL) [17] for a sensor node that is performing a business process. In addition to that, they have introduced a SOAP-compressed message (SCM) technique and a Lean Transfer Protocol (LTP).

Amundson *et al.* [18] have presented a model that can add and access any web service. The authors have assumed that the sensor network application as a well-defined interfaces graph of modular and autonomous service which allows them to be described, published, discovered, and invoked over the network. This provides a convenient way for integrating services from heterogeneous sensor systems.

In addition to the studies related with customizing the XML file, several studies have also focused on customizing the TCP/IP protocol for sensor web services. In the next section, redesign of transport and network protocols will be discussed.

2.2. Redesigning the Transport and Network Protocols

The implementation of web services on the sensor node needs redesign of the transport and network layer protocols such that both the sensor node and the connected device can communicate and transmit messages. An eight bit microcontroller TCP/IP protocol has been presented in [19]. This protocol keeps the same functionality as the conventional TCP/IP, but it minimizes the interface between the TCP/IP stack and the application. In [20], the authors have introduced the spatial IP address assignment scheme for the WSN. This scheme provides a semi-unique IP address to the sensor nodes in order to reduce the protocol overhead.

In [16], the authors have presented a web service design approach. The design approach is flexible, allowing additional nodes to be added to the initial deployment. The authors have proposed several techniques to optimize the TCP/IP protocol to fit the sensor node, which can be summarized as: (i) Persistent TCP connections, (ii) disable delayed TCP acknowledgement, (iii) link layer retransmission, (iv) low power mode between TCP messages and (v) link layer fragmentation. Persistent TCP connections rely on the fact that small number of applications will be using the sensor web services. It has been considered to be more efficient for the client application to maintain an open connection with the sensor node. Hence, the number of messages needed for the TCP/IP transfer is reduced. Disabling delayed TCP acknowledgement modifies the acknowledgement mechanism of the conventional TCP. In the conventional TCP, acknowledgement mechanism works well for the congested network, but in sensor web services, disabling TCP acknowledgement messages reduces the TCP latency. Link layer retransmission is utilized since, even though TCP achieves end-to-end reliability through retransmission,

it can cause a significant delay. Therefore, link-layer Automatic Repeat Request (ARQ) has been used by the authors. Low power mode between TCP messages aims at reducing the energy consumption of the sensor nodes. Keeping sensor radio signals on between packets sending will lead to an energy overhead. To reduce the energy loss, the radio is turned off between TCP packets transmission. Link layer fragmentation is done by segmenting each packet into smaller packets in order to reduce message delivery times and power consumption.

3. Sensor Web Service Applications

In the simplest form, SWS can be used in most applications that employ WSN in order to retrieve data as well as feed it with data to perform specific tasks. One of the domains that utilize SWS is Body Sensor Network (BSN) [21,22], where a set of sensor nodes are deployed on the patient body to monitor the psychological activities and health issues, such as body temperature, motion, sleeping pattern, and heart beats. Here each sensor might host a WS that report certain information to main station in a regular base. Also, SWS can be used in the smart transportation system to monitor highways and roads status and to change the directions of routes remotely. Other applications include precision agriculture, in which SWS are deployed within the farm area to monitor the atmosphere and actuate the water feeding system.

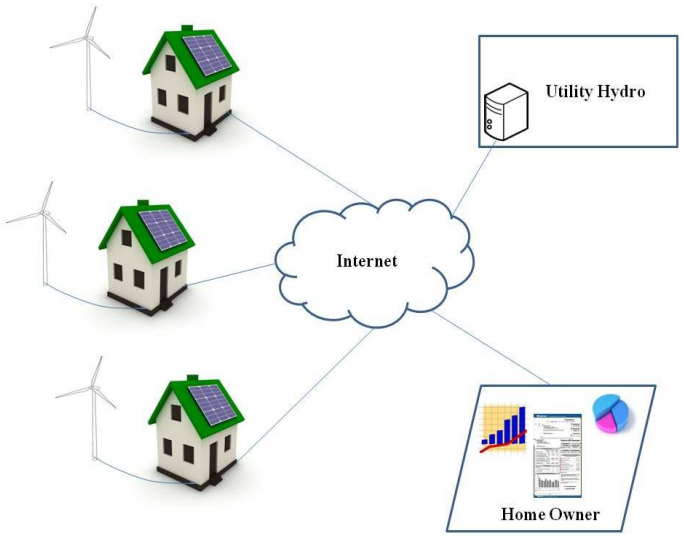
4. Sensor Web Services for the Smart Grid

Sensor web services technology has been used previously in several smart grid applications. Warmer *et al.* [23] have presented the requirements for integrating web services to a smart grid from technical and management perspective. The authors have emphasized the significance of (i) end-user feedback, (ii) automated decentralized control of distributed and demand response, (iii) control of grid stability and islanding operation.

In [24], the authors propose sensor web service applications targeting energy savings and utilization of renewable energy resources in the smart grid. The model has been based on implementing the web service into the sensor node directly, in order to let the utility server communicate directly with the sensor node and control the home appliances.

Figure 1 illustrates the model used for the smart grid applications using sensor web services. The model assumes that the smart home has a smart thermostat and several smart appliances that have sensor nodes to measure their electricity consumption. In addition, a smart meter is available and it is able to collect, measure, and analyze the energy consumption of the smart home. The smart meter is assumed to have the capability to communicate with the utility and consumer devices. For energy generation at home, solar panels and small wind tribunes are considered. Smart grid applications include consumption report, consumption control and management of renewable energy resources. The inhabitants of the smart home can use the locally generated energy instead of drawing power from the smart grid or they can also sell electricity back to the grid. The sensor network is assumed to have a sensor gateway, which is between the WSN and the web server. The utility company and the inhabitants can access this web server. The gateway node can retrieve the information requested by the web server from the sensors that are on the appliances, by accessing the web service in each sensor.

Figure 1. Smart grid applications model using sensor network web services.



The model has been built to support the following applications:

1 Reading Smart Meter Remotely: by allowing utility server to read the meter value along with the time stamp.

2 Control Air Conditioning (AC): through a predefined algorithm, the utility server could decrease the current AC consumption to decrease the load on its generators during the peak hours.

3 Managing Renewable Energy Integration: where the application manages the usage of the energy from two different resources, *i.e.*, the grid or the renewable.

Sensor web services have been shown to provide efficient means of meter reading as well as reducing the consumption of the air conditioners. In addition, managing renewable energy via sensor web services has provided savings for the consumers. Figure 2, presents the amount of the energy saved by running the application for remote appliance control. The application is assumed to run in three different smart homes. In each home, there is a sensor node combined with the AC. The sensor node runs a web service and is able to read the current AC consumption. The energy savings are given in kWh considering five periods of peak hours. The Control Air Conditioning application has shown that savings increase for the regions or seasons that have longer peak hours.

In [25], an application to manage the PHEV charging has been presented. Figure 3 illustrates the system model used to implement the application. The model assumes that the PHEV has a sensor node that hosts a web service to communicate with the car battery and the gas tank. On the other hand, the main application that was developed is located on the utility web server and it is accessed through the internet from an end-user device (PC, cell phone). In order for the application to determine the convenient fuel, it needs to know the electricity price, the peak times, and the gas price. Therefore, the application communicates with the web service located on the utility web server to request the current energy price. Besides, the application accesses a gas price web service to learn the current gas price. This application helps the driver to charge her car or fill it with gas by providing the prices for electricity and gas needed to travel for a certain distance, depending the peak hours. Meanwhile, it reduces the load

on the grid during peak hours. The results showed that the application reduced vehicle driving costs, and it saves a reasonable amount of CO_2 emission.

Figure 2. The amount of energy savings by running the Remote Appliances Control application.

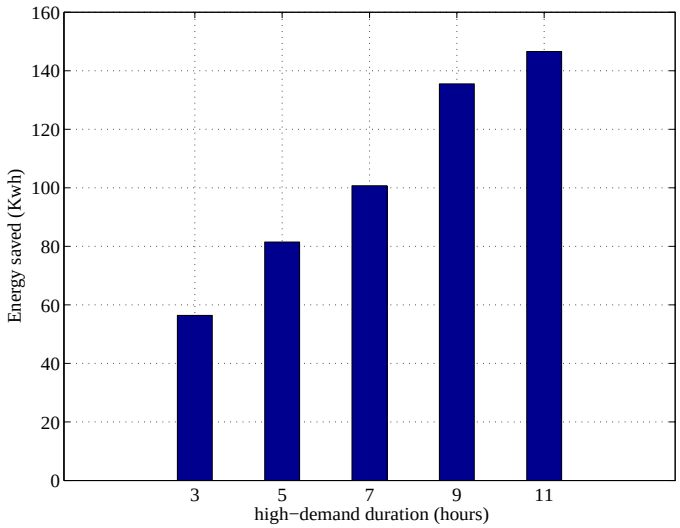
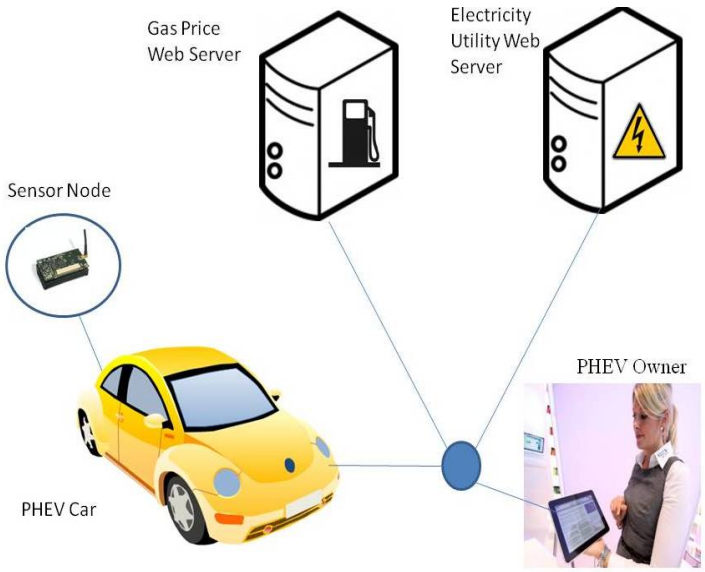


Figure 3. Management of PHEV charging model from the smart grid using sensor web services.



Since the web service code has to be implemented on a sensor node, the sensor characteristics have to be considered in the code design. Therefore, in [24,25], compacting method was used to compress the method names and arguments on the sensor node. In that way, the code sizes and the message sizes have been decreased dramatically. Table 1 represents the method and message sizes after compacting for the mentioned applications.

Table 1. The code & message sizes (Bytes).

Method Name	Code Size	Req. Message Size	Res. Message Size
GtMetRdng	103	218	275
ACRdng	87	275	270
SetAC	124	267	268
MngRen	497	215	269
GtGasRdng	545	295	247

5. Discussion

Implementing WS into individual sensor node needs special attention due to the lack of the node resources—specifically, CPU, memory, and limited energy. Most of the research targeting this area, *i.e.*, developing SWS, is trying to customize the conventional WS components to consume less power, less memory space, and less processing time.

To reduce the node space consumption and shorten the processing time, minimizing the size of the XML file, which is the standard data format to exchange WS messages, is a key point. One procedure is to leverage XML compressor tool to reduce the space complexity of the message, and in turn, reduce data processing time. Another important issue is choosing method and variable's names to be as short as possible (compact strategy), because the WSDL that represents the WS contains all the information about the methods and the parameters, so picking short names in the development phase will produce a WSDL file with less size in addition to less SOAP request/response messages sizes. From an implementation point of view, a system designer can merge more than one XML optimization technique when implementing SWS. For instance, using a compressor tool after following compact strategy will yield a SWS with a smaller size of data, and smaller messages sizes.

In terms of reducing the power consumption, several works have investigated the feasibility of redesigning the network layer. One is to minimize the interference between the network and the upper application layer in term of time and components communication, which yields less power consumption. Another procedure is to reduce the number of messages in the TCP protocol. Such reduction includes getting rid of delayed TCP acknowledgement. Also, on the other hand, keeping persistent TCP connection between the client and the application will reduce the number of messages needed to establish new connections. In general, both procedures fit the sensor capabilities, but the latter is more efficient, since it keeps the number of messages to be as minimal as possible, which in turn saves a considerable amount of energy used to establish new connection.

6. Conclusions

In this survey, we presented the state of art for implementing sensor web services by showing the opportunities and the limitations of the technology. Implementing web services on the sensor nodes has many advantages. First, web services technology is platform-independent. Second, the protocols and data format are easy for the developers to understand and deal with. Third, it is flexible enough to allow the reuse of the service in different systems. On the other hand, implementing web services on the

sensor nodes has several challenges due to the resource constraints of the sensor nodes. To address this problem, many studies have focused on customizing the web services and its components.

In this paper, we have presented a brief survey of the techniques that propose modifications on web services in order to improve the performance of sensor web services. Furthermore, we have presented a case study in an emerging area, *i.e.*, the use of sensor web service technology in smart grid applications. Sensor web services seem to be a promising tool to propagate the data between the smart home and the utility. We have provided a survey on the recent studies on the smart grid applications that utilize sensor web services. We introduced a home application where utilities and users remotely control their smart appliances. In addition, we have presented a Plug-in Hybrid Electric Vehicle (PHEV) application that chooses the most convenient source of fuel depending on the prices.

Sensor web services technology is not limited to the smart grid. Smart health, smart living spaces and finally, smart planet applications can highly benefit from sensor web services.

References

1. Delin, K.; Jackson, S.; Some, R. Sensor webs. *NASA Tech. Briefs* **1999**, *20*, 80.
2. Gibbons, P.; Karp, B.; Ke, Y.; Nath, S.; Seshan, S. Irisnet: An architecture for a worldwide sensor web. *IEEE Pervasive Comput.* **2003**, *4*, 22–33.
3. Moodley, D.; Simonis, I. A New Architecture for the Sensor Web: The Swap Framework. In Proceedings of International Semantic Web Conference, Athens, GA, USA, 5–9 November 2006.
4. Erol-Kantarci, M.; Asad, O.; Mouftah, H. Sensor web services for a smarter planet. *IEICE Trans. Infor. Syst.* **2012**, *E94-D*, 1792-1799.
5. Sahin, D.; Kocak, T.; Ergut, S.; Buccella, C.; Cecati, C.; Hancke, G.P. Smart grid technologies: Communication technologies and standards. *IEEE Trans. Ind. Inform.* **2011**, *7*, 529–539.
6. Gungor, V.; Lu, B.; Hancke, G. Opportunities and challenges of wireless sensor networks in smart grid. *IEEE Trans. Ind. Electron.* **2010**, *57*, 3557–3564.
7. Wang, W.; Xu, Y.; Khanna, M. A survey on the communication architectures in smart grid. *Comput. Netw.* **2011**, *55*, 3604–3629.
8. Glombitza, N.; Pfisterer, D.; Fischer, S. Integrating Wireless Sensor Networks into Web Service-Based Business Processes. In Proceedings of the 4th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks, Urbana Champaign, IL, USA, 30 November–4 December 2009; pp. 25–30.
9. Delin, K.; Jackson, S. The sensor web: A new instrument concept. *Proc. SPIE* **2001**, *4284*, doi: 10.1117/12.426856.
10. Chu, X.; Buyya, R.; Mahalik, N. Service Oriented Sensor Web Block. In *Sensor Network and Configuration: Fundamentals, Standards, Platforms, and Applications*; Springer-Verlag: Berlin, Germany, 2007; pp. 51–74.
11. Woo. A. Demo Abstract: A New Embedded Web Services Approach to Wireless Sensor Networks. In Proceedings of the Fourth ACM SenSys Conference, Boulder, CO, USA, 31 October–3 November 2006; p. 347.

12. Hashmi, N.; Myung, D.; Gaynor, M.; Moulton, S. A Sensor-Based, Web Service-Enabled, Emergency Medical Response System. In Proceedings of the Workshop on End-to-end, Sense-And-Respond Systems, Applications and Services (EESR'05), Berkeley, CA, USA, 5 June 2005; pp. 25–29.
13. Sensor Model Language SML. August 2010. Available online: http://www.botts-inc.net/vast.html (accessed on 20 December 2012).
14. Augeri, C.; Mullins, B.; Baird, B.; Bulutoglu, D.; Baldwin, R. An analysis of XML Compression Efficiency. In Proceedings of ACM Workshop on Experimental Computer Science, San Diego, CA, USA, 13–14 June 2007; Article No. 7.
15. Maneth, S.; Mihaylov, N.; Sakr, S. XML Tree Structure Compression. In Proceedings of 19th International Conference on Database and Expert Systems Application (DEXA'08), Turin, Italy, 1–5 September 2008; pp. 243–247.
16. Priyantha, N.; Kansal, A.; Goraczko, M.; Zhao, F. Tiny Web Service: Design and Implementation of Interoperable and Evolvable Sensor Networks. In Proceedings of the ACM Conference on Embedded Network Sensor Systems, Raleigh, NC, USA, 4–7 November 2008; pp. 253–266.
17. W3C. Bpel Orchestration of Rule-Based Web Services. Available online: http://www.w3.org/2005/rules/wg/wiki/UCR/BPEL (accessed on 20 December 2012).
18. Amundson, I.; Kushwaha, M.; Koutsoukos, X.; Neema, S.; Sztipanovits, J. Efficient Integration of Web Services in Ambient-Aware Sensor Net Applications. In Proceedings of the 3rd International Conference on Broadband Communications, Network and System, San Jose, CA, USA, 1–5 October 2006; doi: 10.1109/BROADNETS.2006.4374298.
19. Dunkels, A. Full TCP/IP for 8-bit Architectures. In Proceedings of the International Conference on Mobile Systems, Applications and Services, New York, NY, USA, March 2003; pp. 85–99.
20. Dunkels, A.; Alonso, J.; Voigt, T. Making TCP/IP Viable for Wireless Sensor Networks. In Proceedings of Work-in-Progress Session of the first European Workshop on Wireless Sensor Networks, Berlin, Germany, January 2004.
21. Ullah, S.; Higgins, H.; Braem, B.; Latre,B.; Blondia, C.; Moerman, I.; Saleem, S.; Rahman, Z.; Kwak, K.; Comprehensive survey of wireless body area networks. *J. Med. Syst.* **2010**, doi:10.1007/s10916-010-9571-3.
22. Chen, M.; Gonzalez, S.; Vasilakos, A.; Cao, H.; Leung, V. Body area networks: A survey. *Mobile Netw. Appl. J.* **2010**, *16*, doi: 10.1007/s11036-010-0260-8.
23. Warmer, C.; Kok, K.; Karnouskos, S.; Weidlich, A.; Nestle, D.; Selzam, P.; Ringelstein, J.; Dimeas, A.; Drenkard, S. Web Services for Integration of Smart Houses in the Smart Grid. In Proceedings of Grid-Interop Conference, Atlanta, GA, USA, 17–19 November 2009; pp. 207–211.
24. Asad, O.; Erol-Kantarci, M.; Mouftah H. Sensor Network Web Services for Demand-Side Energy Management Applications in the Smart Grid. In Proceedings of IEEE Consumer Communications and Networking Conference, Las Vegas, VA, USA, 9–12 January 2011; pp. 1176–1180.

25. Asad, O.; Erol-Kantarci, M.; Mouftah H. Management of PHEV Charging from the Smart Grid Using Sensor Web Services. In Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, Niagara Falls, ON, Canada, 8–11 May 2011; pp. 1246–1249.