Integrating Households into the Smart Grid

Andrea Monacchi*, Dominik Egarter*, Wilfried Elmenreich*†

*Institute of Networked and Embedded Systems,
Alpen-Adria-Universität Klagenfurt / Lakeside Labs, Austria
andrea.monacchi@aau.at, dominik.egarter@aau.at, wilfried.elmenreich@aau.at

†Complex Systems Engineering, University of Passau, Germany
wilfried.elmenreich@uni-passau.de

***Abstract*—The success of the Smart Grid depends on its ability to collect data from heterogeneous sources such as smart meters and smart appliances, as well as the utilization of this information to forecast energy demand and to provide value-added services to users. In our analysis, we discuss requirements for collecting and integrating household data within smart grid applications. We put forward a potential system architecture and report state-of-the-art technologies that can be deployed towards this vision.**

Keywords: smart energy, smart home, smart appliances, semantic sensor networks

I. Introduction

The smart grid is a cyber-physical system combining the electricity infrastructure with an informative channel to connect producers and consumers to become more efficient and reliable. The presence of this data flow will result in modifications to the current energy market, such as new tariff plans based on the actual availability of energy. This aspect is getting more important, as the grid will have to cope with fluctuations produced by a dynamic energy demand and with the integration of renewable-energy plants, which might depend on the availability of sun and wind. Banks of batteries might be used to mitigate such variations, by storing energy when demand is low and a high amount of energy is available. In this way, the use of conventional (i.e., nuclear, fossil fuel) power plants can be kept to a minimal level by relying on them only in case of high energy demand.

Households play an important role in the grid, as users can aim to contribute to an idea of sustainable living and receive tailored services, for instance, appliances can be scheduled to postpone certain tasks and avoid running in periods of peak demand (when the energy might come at a higher price).

Persuasive technologies, such as feedback systems providing real-time energy monitoring and recommendations, have been shown to raise the awareness of users about their energy consumption and inducing a long-term change in their behavior and lifestyle [1]. However, studies show that the effectiveness of these systems in making people responsible depends on their sensitivity and motivation [2]. In addition, intelligent controllers can be used to optimize the running costs of the household, while simultaneously increasing the comfort of inhabitants. This requires taking continuous changes in the grid into account, which is only possible in presence of a reliable and shared data infrastructure.

To achieve this, we argue that openness of protocols and data is necessary to provide the stakeholders of the smart grid with a shared development framework. Indeed, the household is an ecosystem of highly heterogeneous digital devices, using different protocols and data representations. The creation of an advanced metering infrastructure that is able to collect and analyze energy consumption information down to the device level is expected to extend such applications to the global market.

In this paper we advocate for the integration of data produced within domestic environments into the smart grid. Our contribution is twofold: presenting design guidelines and identifying technologies that can be deployed towards this vision. In Section 2, we outline potential requirements for a data infrastructure that can mediate the interaction between the stakeholders of the smart grid, such as users in domestic environments, energy providers and developers of applications. To effectively implement the architecture, the paper proposes potential solutions which emerge from current research trends. In particular, we identify open protocols that can enhance embedded devices with networking capabilities. Section 3 regards current solutions to integrate devices into the Internet. In addition, we propose using techniques from the semantic web for allowing devices to describe their data. Section 4 suggests ways of annotating data and devices using the Resource Description Framework (RDF) model, as well as mechanisms to retrieve descriptions in a wide scale network.

II. Requirements

A reliable and flexible infrastructure able to tackle different demand is required to handle the amount of data produced by heterogeneous devices in highly dynamic environments. We have identified the following requirements:

- **Plug & play mechanism:** The architecture should support discoverability of services and resources in the network so that they can be used as soon as they become available [3]. Services are required to provide a description of their characteristics that can be advertised to other peers or retrieved when needed.
- **Accessibility of data:** To increase interoperability and reduce maintainance costs, the architecture should support a data-driven abstraction. This data space should be accessible through a uniform interface, such as standard query languages and a well defined API. This requires sensed data to be semantically annotated

according to well-known design patterns and vocabularies, and enhanced by situational information such as time and space.

- **Reliable and neutral data infrastructure:** Knowledge and information about properties and context should be stored in a repository that can ensure availability and continuity of service. Moreover, it should support integrity of data and avoid any discrimination that is not strictly required to guarantee Quality of Service.
- **Confidentiality:** The architecture should provide mechanisms to avoid unauthorized disclosure of information. In particular, it should secure access to the home network and the repository (e.g., using authentication and encrypted communication) and use a sandbox mechanism for applications when accessing user data (e.g., OAuth protocol[1]).
- **Quality of data:** The architecture should ensure appropriateness of data, i.e., consistence with respect to time. For instance, event-oriented systems may not be able to meet strong real-time constraints, as events are queued for an unpredictable time before being dispatched.

A. A potential architecture

A potential architecture addressing all the requirements should firstly support openness of data and protocols. The household can be considered as a network of self-describing sensors and actuators, that can dynamically join and leave the network with the help of service discovery mechanisms. Such devices are producers and consumers of data. Therefore, the household can be seen as data space, where data are described with respect to shared vocabularies and well-known design patterns, and can be retrieved using standard query languages. For instance, data can be produced by meters, appliances and generators and related to situational information such as time and space. A smart appliance is aware of its consumed power, based on voltage and current measurements or built-in profiles [4]. This requires a machine-readable description to be deployed with the smart device to describe its properties and functionalities, so that they can be automatically used by other machines. In addition, such description can also be used as a reference for describing data produced by the appliance. For instance, a washing machine might be reachable at a certain URI (Uniform Resource Identifier) and return metadata describing the device, such as profile listing the type and the expected consumption for a certain task. The appliance may provide various features, such as measuring its actual consumption and returning it as a stream annotated with respect to the device profile.

A gateway is used to bridge the home network to the Internet, and to ensure security when accessing home devices. Moreover, it plays also a crucial role in the integration of smart and legacy devices, which can not provide their metadata, and thus, would not be accessible within this architecture. Non-intrusive load monitoring (NILM)[5] can be applied to disaggregate the power profile of attached loads out of the overall household consumption. Accordingly, metadata can be dynamically associated to the power profile of running appliances in order to be directly exploited by applications.

The data space should not be localized to the house, as such data can be useful to build a more accurate understanding of the system and offer tailored functionalities, which can take dynamic variations of the market and the demand into account. We look at cloud-computing technologies to cope with the huge amount of data and computing power required to manage the repository. The development of batch and real-time analytics solutions, such as Hadoop[2] and Storm[3] respectively, enables the stakeholders of the smart grid to perform complex data analysis, such as forecasting of future demand. Therefore, future home management systems will be able not only to optimize energy consumption but also to track the inhabitants' activities, and offer complex tailored functionalities. Rule-based semantic reasoning can be used to infer further knowledge and produce more abstract situational information that can be directly exploited by decision makers implemented as applications. This will provide users a set of value-added services and help data providers pay off the operational costs of the repository. A sketch of our architecture is shown in Fig. 1. In the following sections we describe in greater detail potential technologies that could be employed to implement this architecture, in terms of communication infrastructure and ways of semantically annotating data and resources so that they can automatically be consumed in a network of heterogeneous entities.

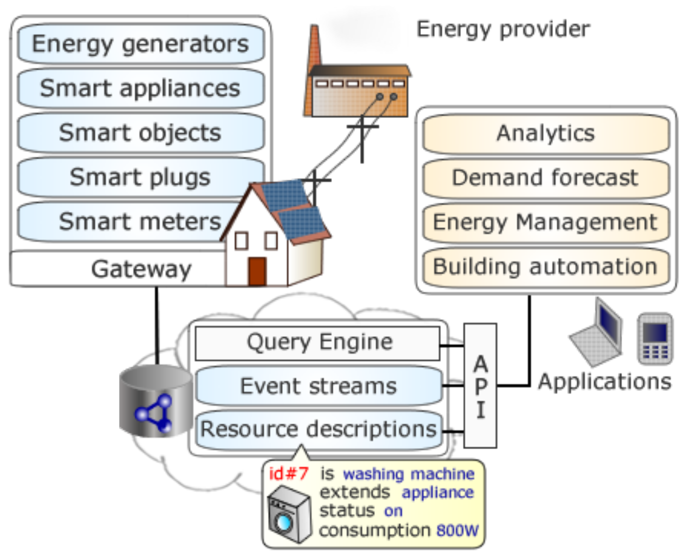

Fig. 1. Architecture to provide data interoperability in the smart grid: data produced in households is semantically annotated and fed to the cloud. A query engine provides a uniform interface to data that can be exploited by applications.

III. The communication infrastructure

As shown in [6], a web-enabled smart grid can be realized by deploying inexpensive devices to monitor energy-related aspects such as power quality. Bringing the web into

[1] http://oauth.net/

[2] http://hadoop.apache.org/

[3] http://storm-project.net/

constrained networks requires embedded devices to use a lightweight and interoperable stack of protocols that is fully compatible with the existing Internet. This is already possible, as some lightweight versions targeting embedded systems have already been implemented. The constrained application protocol (CoAP)[4] has been proposed as alternative to HTTP that can be run by deeply embedded systems (e.g., 8-bit microcontrollers). It uses UDP with a simple retransmission mechanism where each GET request is associated with a unique identifier. In addition, it provides an eventing mechanism that allows subscription to updates of a certain resource. In the REST model [7], resource discovery takes place by accessing an index page listing links leading to resources on the same or different servers. Therefore, it may be argued that service-oriented architectures such as the DPWS[5] are more powerful than classical REST designs, especially in terms of service discovery. CoAP tries to remedy this problem by using the CoRE link format (RFC6690)[6], where resources can be explained by meaningful links. Since links define relationships between web resources, assigning a meaning to links enables machines to automatically calculate how to use resources on a server. In particular, a client can contact the server using a GET to a predefined location, i.e., /.well-known/core, which returns the list of resources exposed by the server and their media type. This means that nodes can act as resource directories listing links to resources stored on other nodes. To be visible to other clients, resource providers can POST a link to their resources to the "/.well-known/core" position of the chosen directory node. For instance, a washing machine could post its description on the gateway, in order to be visible within the network. An energy management system might need to retrieve the current consumption of the appliance by performing a GET to the sensing resource of the node "washing machine" (e.g., *coap://DNS-name-or-IPv6-address-of-washing-machine/consumption-sensor*). Similarly, the application could act on the appliance by using POST to modify the state of a certain resource. For instance, the system could update the information of the current energy cost to enable localized decision-making tasks, as well as directly controlling the device (e.g., switching it on/off) when certain situations occur. CoAP has been implemented in different programming languages: C (libCoap[7][8] and Erbium [8][9]), C++ (evCoap[9]), Java (Californium[10], JCoAP[11]), Javascript (Copper[12]). Along with 6LoWPAN[13], CoAP is the ideal candidate to integrate real world services into the standard internet network (Fig. 2). Therefore, web servers can be run on embedded devices (e.g., washing machine) and used to interact with humans (i.e., providing a service), as well as machines (i.e., returning machine-readable information). Security in real world objects is also crucial. A secure CoAP (CoAPs) using a compressed DTLS over 6LoWPAN networks is discussed in [10].

[4]http://tools.ietf.org/html/draft-ietf-core-coap-12
[5]http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01
[6]http://tools.ietf.org/html/rfc6690
[7]http://sourceforge.net/projects/libcoap/
[8]http://www.contiki-os.org/
[9]https://github.com/koanlogic/webthings/tree/master/bridge/sw/lib/evcoap
[10]https://github.com/mkovatsc/Californium
[11]http://code.google.com/p/jcoap/
[12]https://github.com/mkovatsc/Copper
[13]http://tools.ietf.org/wg/6lowpan/

Fig. 2. The HTTP-CoAP mapping: integrating embedded systems in the internet

A. Reducing the data representation overhead

Data interoperability in the interaction is ensured by standard data-interchange formats, such as XML and the lighter Javascript Object Notation (JSON). Binary variants have been proposed to reduce the overhead of managing these formats in deeply constrained embedded devices. The Efficient XML interchange (EXI) [14] provides an efficient way of processing XML information. It is shown to provide the highest compression and compactness compared with respect to other representations [11]. An implementation is proposed by [12]. Its counterpart is the binary JSON format[15].

B. A CoAP gateway for the home

In our proposed architecture, the gateway acts as sink and resource directory for the network. Networked devices, such as smart appliances, can interact with the gateway by registering themselves and providing their description when necessary. The *if* (interface description) attribute of the CoRE Link format can be used to refer to an external machine-readable document describing the resource. The description is usually specified in formats such as WSDL or WADL, which can only capture aspects related to the interface. However, developing applications as workflows of CoAP web services requires human involvement. According to [13], machines should be able to figure out which operations can be executed on resources, and which effects operations produce in a certain situation. In this way, they can use a uniform interface to retrieve descriptions and autonomously manipulate resources according to a certain goal. RESTdesc[16] is a mechanism according to which RESTful webservices can be described in terms of pre- and post-conditions in the Notation-3[17] format. For instance, it can be used to define the effects of performing a POST method, which is strictly dependent on the application logic and the data passed with the request. In [14], RESTdesc is used to show the composition of sensor web services for the reservation of tables in a restaurant. However, to truly integrate household appliances and devices within the household, and consequently in the smart grid, the gateway will have to mediate between smart and legacy devices. Therefore, it will have to connect to a sensing system in order to detect running devices. On one hand, smart outlets can be connected to running devices to track their power consumption. On the other hand, non-intrusive load monitoring (NILM) [5] exploits optimization [15] and machine-learning techniques to disaggregate the power profile of loads out of the overall household consumption. For instance, a refrigerator can be identified by

[14]http://www.w3.org/TR/exi/
[15]http://bjson.org/
[16]http://restdesc.org/
[17]A RDF serialization format. http://www.w3.org/TeamSubmission/n3/

its periodic power profile, whereas a certain sequence of state changes can help detect a multi-state appliance, such as a coffee machine. Detected appliances can be represented on the gateway as virtual resources, returning their actual and expected consumption. This requires the definition of resource profiles, which can be dynamically filled out by the gateway to represent these resources. In this way, the architecture is capable of collecting energy consumption information from any device in the household. Consequently, applications can already exploit this information, without waiting for the market to provide effective development frameworks for producers of devices.

IV. ENABLING DATA INTEROPERABILITY

Linked data is a paradigm which aims at leveraging the web from a collection of documents to a network of interlinked data called the Web of Data. Beside the web of hyperlinked HTML (HyperText Markup Language) documents, Linked Data suggests the use of RDF (Resource Description Framework) for describing the content of documents, in terms of relationships between data. In the RDF data model, the basic unit of information can be described as *subject*, *predicate* and *object* triple. A predicate is a property of the subject entity and provides a connection to another entity (object) or a literal value (e.g., string). This means that various triples denote a directed graph, where nodes can represent real-world entities (e.g., people, places) or abstract concepts. In order to avoid naming ambiguities, RDF uniquely assigns URIs to resources. Since many URIs are also URLs (i.e., the set of names used to identify web resources), many people erroneously use them interchangeably although they do not denote the same set of resources (i.e., URLs $\subset$ URIs). Links can be used to specify properties of certain resources, by providing connections to vocabularies written in the RDFS (RDF Vocabulary Definition Language) and OWL (Web Ontology Language) formats. In addition, links can also refer to entities residing on other data sources, so that the web of data can be navigated by humans and crawlers following RDF links. This is a clear advantage over other kinds of web services, as rather than using customized mashups built from statically chosen data sources, data can be directly discovered and integrated in the workflow without human intervention [16]. We refer to [17] for a thorough introduction to linked data.

Ontologies can be used as agreed vocabularies between data consumers and data producers, as ontologies represent knowledge in a certain domain. Therefore, data can be semantically annotated with respect to concepts defined in such vocabularies. For example, a description of a washing machine could be defined in Turtle[18] as:

```
@prefix ns: <http://myrepository.com/houses/12345/> .
@prefix en: <http://example.com/ontologies/appliances.owl#> .

ns:washing-machine en:model         "XYZ123456" ;
                   en:manufacturer  "Bob Inc." ;
                   en:type          en:washing-machine ;
                   en:consumption   "800" .
```

This could be deployed with the appliance and retrieved at the URI "http://myrepository.com/houses/12345/washing-machine", which might link to the actual IPv6 of the washing machine. A common strategy[19] is to retrieve resources using their URI and exploit headers to specify whether retrieving a web page or an RDF description. In this way, humans (via a web browser) and machines can access two different representations of the same resource, identified by its URI. In HTTP, the URI of the representation is returned to the client using the code 303 (i.e., see others). The client can then retrieve it through a common GET request. A straightforward way to exploit this semantic description would be to dereference URIs and following RDF links to navigate the graph and discover other resources on the fly. For instance, in the proposed architecture the URI of the gateway could be an informative point to discover resources in the household. However, this might be a slow process when done in a large scale network, such as the Web. A common solution used in search engines is to navigate the graph off-line using a crawler. Collected data is processed and stored in a persistent storage such as a triple store. In our architecture, smart devices maintain a description of their capabilities and properties that can be retrieved at a certain URI, as well as streams of linked sensor data that refer to such descriptions. However, to provide applications with an efficient way to retrieve stream data from the network of constrained devices, we collect sensor data in a reliable data repository (i.e., a triple store) and we provide an endpoint where queries can be executed. In fact, managing heavyweight knowledge representation techniques on resource constrained devices might not be possible, as they would produce obsolete results in highly dynamic environments. The introduction of lightweight semantic tools and models, such as the binary RDF[20] representation, is therefore required to bring semantic technologies to these networks.

A. Towards a widely accepted ontological framework

The definition and agreement of common vocabularies is crucial to extend an ontological framework to a global scale, which means that everyone sharing a certain vocabulary will be able to interpret information. Indeed, data might use different vocabularies to represent the same concept. Therefore, to provide a homogeneous annotation of data it is important to converge to a single vocabulary (e.g., translating data coming from different data sources), as well as defining correspondences between concepts defined in different vocabularies (i.e., ontology alignment). The W3C Semantic Sensor Network Incubator group addresses this problem by providing a standardized vocabulary. The SSN-XG ontology[21] is an OWL ontology that describes sensors in terms of accuracy and capabilities, as well as observations that can be drawn from the recorded data [18]. The ontology is designed according to the stimulus-sensor-observation design pattern, which separates environment stimuli from the kind of sensor used for the data collection [19]. This allows the reuse of stimuli, sensors and observations in different contexts. Information on how to enrich sensor data within the SSN-XG ontological framework is shown in [20]. In particular, they provide information about the type of data collected: where, when and under which conditions. A specific ontology for capturing event data in smart grids is proposed in [21], whereas [22] introduces the

[18]Terse RDF Triple Language, a RDF serialization format. http://www.w3.org/TeamSubmission/turtle/

[19]http://www.w3.org/TR/cooluris/

[20]http://www.w3.org/Submission/2011/03/

[21]http://purl.oclc.org/NET/ssnx/ssn

Smart Appliances Ontology Model, an ontological framework for smart appliances.

B. Querying the web of sensor data

Linked data is described as set of RDF triples denoting a labeled graph. The possibility of retrieving data from such graphs by expressing complex queries across diverse data sets is therefore the key for automatic composition of data and services. SPARQL[22] (SPARQL Protocol and RDF Query Language) is the most diffuse query language for retrieving data defined in the RDF format. Starting from version 1.1, SPARQL provides the possibility of adding, deleting and changing data triples. These kinds of features are expected to provide enough flexibility to RDF definitions. For instance, applications might require updates of certain properties of running devices, as well as addition and removal of certain relationships between data to encode certain rules in the knowledge base. For this reason, [23] foresee the evolution of the Web to a read-write space, where machines can collaborate. In this scenario, potential concerns related to the provenance and trust of this data arise, as well as the permission to manipulate them in applications.

As shown in [24], SPARQL can also be used to perform complex event processing over RDF data. A mechanism to use SPARQL for defining rules and constraints on semantic-web graphs is SPIN[23] (SPARQL Inferencing Notation). Accordingly, data constraints can be verified using SPARQL ASK, while SPARQL CONSTRUCT and UPDATE can be used to create new data triples when certain conditions occur. This might be used to perform reasoning tasks as alternative to specific languages (e.g., RIF) and rule engines. New triples might be added as result of RDFS and OWL inferencing, as well as business rules and computations.

SPARQL works effectively in the web of linked data where information changes infrequently and queries produce the same results for a certain time window. However, this is not appropriate for the real world – a source of continuously changing data. Thus a different query language able to capture this dynamism is needed. Several other alternative query languages have been proposed, such as C-SPARQL [25], SPARQLstream [26], EP-SPARQL [27], and CQELS [28]. In the *Linked Stream Middleware* [29] data triples from over 110 000 sensors are collected, and the CQELS language handles both queries on time independent properties and streams of sensor data. The possibility to exploit stream data collected in real environments plays a crucial role in cyber-physical systems such as the smart grid. To this extent the *Super Stream Collider* [30] provides tools to create complex mashups out of linked streams and linked data.

V. Related work

An architecture for a Semantic Web of Things is proposed in [31]. The authors developed an ontology out of concepts defined in existing ontologies, such as in the SSN-XG ontology. In addition, they implemented the architecture using the Jena[24] semantic web framework. A crawler periodically scans the CoAP network and stores sensor metadata in a centralised triple store. The data can be queried using SPARQL, although sensor data streams are not stored into the triple store and can not be queried. The authors refer to future work for this feature, perhaps exploiting a cloud-based infrastructure. Moreover, semantic entities are proposed as solution to map sensor data to their high-level state. In particular, entities are implemented as virtual sensors that expose a RESTful interface to manage the high-level state. In our proposed architecture, we advocate for the completeness of SPARQL for data management and complex event processing. [32] aims at improving the approach presented in [31]. The authors show that the classic CoAP resource discovery can provide a syntatic matchmaking mechanism. Thus, semantic matchmaking is provided by extending the set of standard CoRE attributes with properties that semantically describe the resource (e.g., latitude and longitude). In our architecture, information describing the resource is defined in the resource description file, rather than as CoRE attributes. As the description consists of RDF statements, it can be stored in the triple store along with the context data, and directly queried using SPARQL.

VI. Conclusion

This paper outlines a software architecture for integrating homes into the smart grid, with particular focus on data interoperability. We have discussed architectural requirements to be met when dealing with such scenarios and identified potential technologies that could be employed towards this vision. Several companies such as Bidgely[25], EnergyHub[26] and Ecofactor[27] are already proposing complete solutions to collect and analyse energy consumption in households. Data mining and disaggregation techniques (i.e., NILM) are used to provide users with direct and indirect feedback, such as current and predicted costs, as well as information to help and engage them. However, such systems are specifically built for data sources and applications handling energy consumption data. We expect future real world data repositories to be better integrable with other domains data sets (e.g., social networks, weather channels). For instance, the presence of a SPARQL endpoint on each dataset would allow applications to run federated queries. This would facilitate the exploitation of this collective knowledge for producing more accurate data analytics and value-added tailored services. An immediate solution might be to use a cloud-based triple store such as Dydra[28], which exposes data through a SPARQL endpoint and promises flexibility and scalability, as well as basic authentication features. However, we expect further steps towards the standardization of the semantic framework to allow stakeholders the annotation of data using well-defined patterns and vocabularies. In addition, advances in non-intrusive load monitoring will enhance legacy devices with a basic semantic description, which will make disaggregated information accessible within the architecture. In this way, applications can exploit this information without waiting for the market to provide effective development frameworks for producers of devices. In our future research we will try to bridge the gap between producers and consumers of data in the smart grid by developing concrete solutions to facilitate data integration.

[22] http://www.w3.org/TR/sparql11-query/

[23] http://www.w3.org/Submission/spin-overview/

[24] http://jena.apache.org/

[25] http://www.bidgely.com/

[26] http://www.energyhub.com/

[27] http://www.ecofactor.com/

[28] http://dydra.com

VII. Acknowledgements

This work is supported by Lakeside Labs, Klagenfurt, Austria and funded by the European Regional Development Fund (ERDF) and the Carinthian Economic Promotion Fund (KWF) under grant KWF 20214 — 23743 — 35470. We would like to thank Kornelia Lienbacher and Lizzie Dawes for proofreading the paper.

References

[1] S. Darby, "The effectiveness of feedback on energy consumption: a review for DEFRA of the literature on metering, billing and direct displays," Environmental Change Institute, University of Oxford, Tech. Rep., 2006.

[2] Y. A. Strengers, "Designing eco-feedback systems for everyday life," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 2135–2144.

[3] S. Pitzek and W. Elmenreich, "Plug-and-play: Bridging the semantic gap between application and transducers," in *Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation*. IEEE, 2005, pp. 799–806.

[4] W. Elmenreich and D. Egarter, "Design guidelines for smart appliances," in *Proceedings of the Tenth Workshop on Intelligent Solutions in Embedded Systems (WISES)*, 2012, pp. 76 –82.

[5] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76 –84, 2011.

[6] N. Bui, A. Castellani, P. Casari, and M. Zorzi, "The internet of energy: a web-enabled smart grid system," *IEEE Network*, vol. 26, no. 4, pp. 39–45, 2012.

[7] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[8] K. Kuladinithi, O. Bergmann, T. P. M. Becker, and C. Görg, "Implementation of coap and its application in transport logistics," in *Proceedings of the Workshop on Extending the Internet to Low power and Lossy Networks*, 2011.

[9] M. Kovatsch, S. Duquennoy, and A. Dunkels, "A low-power CoAP for Contiki," in *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2011)*, 2011.

[10] S. Raza, D. Trabalza, and T. Voigt, "6LoWPAN compressed DTLS for COAP," in *Proceedings of 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE Computer Society, 2012, pp. 287–289.

[11] S. Sakr, "XML compression techniques: A survey and comparison," *Journal of Computer and System Sciences*, vol. 75, no. 5, pp. 303–322, 2009.

[12] R. Kyusakov, "Towards application of service oriented architecture in wireless sensor networks," Ph.D. dissertation, Luleå University of technology, 2012.

[13] R. Verborgh, E. Mannens, and R. Van de Walle, "The rise of the Web for Agents," in *Proceedings of the The First International Conference on Building and Exploring Web Based Environments*, Jan. 2013, pp. 69–74.

[14] R. Verborgh, V. Haerinck, T. Steiner, D. Van Deursen, S. Van Hoecke, J. De Roo, R. Van de Walle, and J. Gabarró Vallés, "Functional composition of sensor Web APIs," in *Proceedings of the 5th International Workshop on Semantic Sensor Networks*, Nov. 2012.

[15] D. Egarter, A. Sobe, and W. Elmenreich, "Evolving non-intrusive load monitoring," in *Proceedings of 16th European Conference on the Applications of Evolutionary Computation*. Springer, 2012, to appear.

[16] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, no. 3, pp. 1–22, 2009.

[17] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.

[18] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, no. 0, 2012.

[19] K. Janowicz and M. Compton, "The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology," in *3rd International Workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010)*, 2010.

[20] J.-P. Calbimonte, H. Jeung, O. Corcho, and K. Aberer, "Semantic sensor data search in a large-scale federated sensor network," in *4th International Workshop on Semantic Sensor Networks. Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, 2011.

[21] Q. Zhou, S. Natarajan, Y. Simmhan, and V. Prasanna, "Semantic information modeling for emerging applications in smart grid," in *Proceedings of the 2012 Ninth International Conference on Information Technology - New Generations*. IEEE Computer Society, 2012, pp. 775–782.

[22] Y. Zhnag, Z. Wei, Y. Yang, and C. Song, "Ontology description of smart home appliance based on semantic web," in *Proceedings of International on Conference Computer Science Service System (CSSS)*, 2012, pp. 695–698.

[23] S. Coppens, R. Verborgh, M. Vander Sande, D. Van Deursen, E. Mannens, and R. Van de Walle, "A truly Read-Write Web for machines as the next-generation Web?" in *Proceedings of the SW2012 workshop: What will the Semantic Web look like 10years from now?*, nov 2012.

[24] M. Rinne, H. Abdullah, S. Törmä, and E. Nuutila, "Processing heterogeneous rdf events with standing sparql update rules," in *OTM Conferences (2)*, ser. Lecture Notes in Computer Science, R. Meersman, H. Panetto, T. S. Dillon, S. Rinderle-Ma, P. Dadam, X. Zhou, S. Pearson, A. Ferscha, S. Bergamaschi, and I. F. Cruz, Eds., vol. 7566. Springer, 2012, pp. 797–806.

[25] D. F. Barbieri, D. Braga, S. Ceri, and M. Grossniklaus, "An execution environment for C-SPARQL queries," in *Proceedings of the 13th International Conference on Extending Database Technology*. ACM, 2010, pp. 441–452.

[26] J.-P. Calbimonte, O. Corcho, and A. J. G. Gray, "Enabling ontology-based access to streaming data sources," in *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I*. Springer-Verlag, 2010, pp. 96–111.

[27] D. Anicic, P. Fodor, S. Rudolph, and N. Stojanovic, "EP-SPARQL: a unified language for event processing and stream reasoning," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 635–644.

[28] D. Le-Phuoc, M. Dao-Tran, J. X. Parreira, and M. Hauswirth, "A native and adaptive approach for unified processing of linked streams and linked data," in *Proceedings of the 10th international conference on The semantic web - Volume Part I*. Springer-Verlag, 2011, pp. 370–388.

[29] D. Le-Phuoc, H. N. M. Quoc, J. X. Parreira, and M. Hauswirth, "The linked sensor middleware - connecting the real world and the semantic web," in *Semantic Web Challenge 2011 of 10th International Conference on Semantic Web (ISWC) 2011*, 2011.

[30] H. N. M. Quoc, M. Serrano, D. L. Phuoc, and M. Hauswirth, "Super stream collider–linked stream mashups for everyone," in *Proceedings of the Semantic Web Challenge co-located with ISWC2012*, Boston, US, November 2012.

[31] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kroller, M. Pagel, M. Hauswirth, M. Karnstedt, M. Leggieri, A. Passant, and R. Richardson, "Spitfire: toward a semantic web of things," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 40 –48, nov 2011.

[32] M. Ruta, F. Scioscia, G. Loseto, F. Gramegna, A. Pinto, S. Ieva, and E. Di Sciascio, "A logic-based coap extension for resource discovery in semantic sensor networks," in *5th International Workshop on Semantic Sensor Networks. Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*, vol. 904, nov 2012, pp. 17–32.