Xiang Su, Jukka Riekki, Janne Haverinen, Johanna Nieminen , Jukka K. Nurminen

ENABLING SEMANTICS FOR THE INTERNET OF THINGS – DATA REPRESENTATIONS AND ENERGY CONSUMPTIONS

The development of Internet of Things (IoT) applications can be facilitated by encoding the meaning of the data in the messages sent by IoT nodes, but the constrained resources of these nodes challenge the common Semantic Web solutions for doing this.

Internet of Things (IoT) is expected to bring the Internet truly into our everyday lives by connecting a vast amount of devices and objects (the so-called things) to the Internet. All these things will communicate with other peers and servers in the Internet. The resulting uniform access to things will introduce significant possibilities for IoT applications.

Even more can be achieved if semantics is included in the information produced by the IoT nodes. Semantics enables machine-interpretable and self-descriptive data and facilitates information integration and share, and inference for new knowledge. However, since IoT nodes are often small devices with modest computing, communication, memory and energy resources, they introduce challenges not present in the common scenarios of Semantic Web. Hence, the main challenge is to add semantics without breaking the constraints on resource usage. In this article, we study how to enable richer semantics for IoT data, and evaluate different approaches with energy efficiency with a simple sensor system. Our sensor node measures acceleration and magnetic field, both in three dimensions, and temperature as well. This kind of sensors could be widely deployed in the IoT smart environments. We focus on different data formats enabling semantics, rather than protocols, architectures or ontologies in this paper.

Data formats

One of the main challenges of IoT data formats is mapping between data formats and models used for constrained IoT nodes and data formats and models used in the Web and Semantic Web. A data format should set minimal requirements for both IoT nodes and the consumers of data. That is, the solution should increase the nodes' resource consumption as little as possible, the solution should be general and any consumer should be able to interpret the data with minimal effort and apriori knowledge. Moreover, the data format should be compatible with Semantic Web, as only then the existing Semantic Web tools can be used.

Semantic Web communities, like W3C, have established specifications for formal knowledge representations, like RDF, OWL, N3 and Turtle. These knowledge representations can also be utilized for representing IoT data. The simplest way of semantically representing a measurement made by an IoT device with RDF, is denoting the IoT device as the subject, the measured quantity as the property, and the measured value as the object. For example, "Sensor 1" is the subject, "Temperature" is the property, and "25" is the value. The unit of measurement can be defined separately.

However, these formats are designed to be used by Web applications; hence resource usage was not the main issue in their development. SenML and Entity Notation (EN) [1] are targeted for resource-constrained devices. A SenML description carries a single base object consisting of attributes and an array of entries. Each entry, in turn, consists of attributes such as a unique identifier for the sensor, the time the measurement was made, and the current value. SenML can be represented in JSON, XML and Efficient XML Interchange (EXI) formats. The SenML format can be extended with further semantic custom attributes. For example, the Resource Type attribute can be used to define the meaning of a resource. EN is another lightweight data format that supports Semantic Web technologies. EN has been designed to be compatible with RDF and OWL and it has almost equal expressivity as RDF and N3 on the data exchange level. Its compact format can only include a UUID and some variables (for example, sensor measurements, etc. are variables in EN).

We compare the semantic expressivity of RDF, N3, SenML and EN in Table 1. RDF, N3 and EN can be mapped to conceptual graphs straightforwardly, as they all have a (subject, property, object) triplet structure as the base representation. Hence, they support ontologies. SenML has a more arbitrary data structure, which cannot be mapped to a conceptual graph in a similar fashion. Hence, SenML data cannot be utilized by knowledge-based systems as easily as the other alternatives. On the other hand, SenML may be easy to produce by IoT nodes, because it resembles the basic data structures of programming languages. The compact EN format has the same benefit. The type of the data can be defined with all these formats, which facilitates associating measured data values to concepts. RDF and N3 support rich XML Schema data types, while SenML allows only four basic data types. EN packets do not include data type information, but such information can be accessed from related knowledge representations. All these data formats support external semantic information, but in different fashions.

Figure 1. *System Architecture*

	RDF	N3	SenML	EN
Conceptual Graphs	Y	Y	N	Y
Triplet Relations	Y	Y	N	Y
Device Type	Y	Y	Y	Y
Data Types	XSD	XSD	4 types	N
External Semantics	Y	Y	Y	Y

Table 1. *Data format comparison*

Energy efficiency

Energy consumption is a key issue for IoT nodes. Hence, when semantics is added into IoT, energy-efficiency is a key criterion for comparing alternative solutions. Energy consumption together with other limited resources is one of the key drivers in wireless sensor network research. For example, it is reported in [2] that communication is over 1,000 times more expensive in terms of energy than performing a trivial aggregation operation. However, widely cited surveys [3, 4] do not have any explicit discussion on adding semantics to the data. It seems that integrating sensors into Semantic Web has not yet attracted the attention of researchers.

We measured the energy consumptions of encoding and decoding for different semantic data formats of the same data in a sensor system. As shown in Figure 1, this system consists of two sensors (based on Atmel's 8-bit ATmega32 microcontroller) communicating with Bluetooth and a knowledge processing component on a PC. Sensor A encodes the different formats and sends them to Sensor B. Sensor B decodes these data formats to formats compatible with a knowledge processing component. As a result, the knowledge system can reason additional knowledge and actions based on the data generated by IoT nodes.

Figure 2 presents energy consumption comparison on sensor A. Generating SenML/EXI messages requires more computing energy than other alternatives, but transmission energy consumption for SenML/EXI is among the lowest ones. When comparing overall energy consumption, SenML/EXI requires more energy than the two times longer SenML/JSON and SenML/XML messages. The short EN format requires the least energy and other alternatives consume at least double that amount. Generating short EN messages only consumes about 35% of generating RDF/XML messages, which consume the largest amount of energy. But on the other hand, the receiver of the short EN messages needs one more step (on sensor B or PC) to extend the short EN packet into a complete EN packet that is directly comparable with RDF and N3.

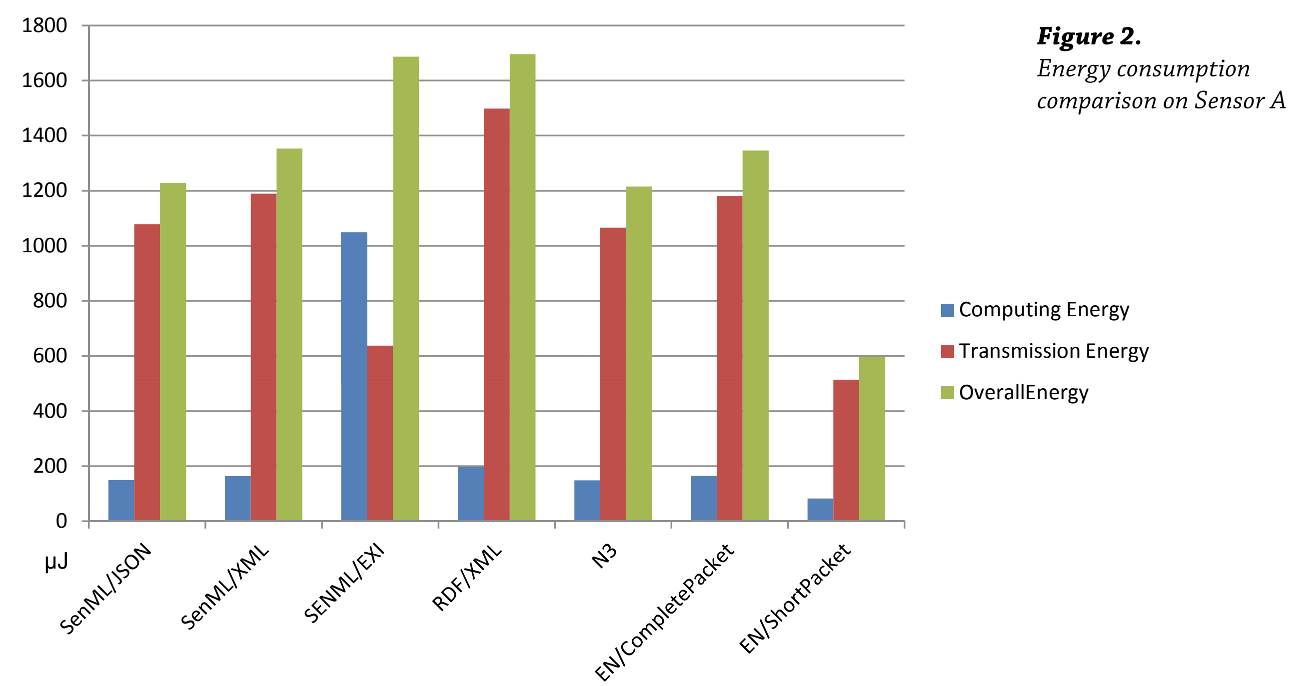

Figure 2. *Energy consumption comparison on Sensor A*

Discussion

We are studying the best ways to add semantics to IoT data. Even this simple experiment illustrates how big a difference a data format can make in energy consumption. One interesting potential scenario for our future work is a gateway receiving data from several similar sensors, aggregating the data values, and sending the resulting data forward.

Many other factors have an effect on energy consumption, but we will mainly focus on data formats supporting semantics; on their expressivity and resource consumption. The other factors include the header lengths of the protocols, messaging patterns and architecture. In addition, the meaning encoded in the messages needs to be shared by all entities producing and consuming the data. That is, ontologies are needed. Moreover, as IoT systems will produce large amounts of data, reasoning techniques that scale and infer useful information in a reasonable amount of time are called for. These reasoning techniques need to be deployable into devices with varying computing resources.

Acknowledgment

This work was funded by the Internet of Things (IoT) program funded by TIVIT and Tekes. We also would like to thank HPY Research Foundation and Tauno Tönning Säätiö for funding. The majority of Janne Haverinen's work was done when he was working at University of Oulu. All of Johanna Nieminen's work was done when she was working at Nokia Research Center, Helsinki.

References

[1] X. Su, J. Riekki and J. Haverinen, Entity Notation: enabling knowledge representations for resource-constrained sensors, Personal and Ubiquitous Computing, volume 16 issue 7, Oct. 2012 pp 819-834.

[2] V. Cantoni , L. Lombardi, P. Lombardi , Challenges for Data Mining in Distributed Sensor Networks, 18th International Conference on Pattern Recognition (ICPR'06), p. 1000-1007 (2006).

[3] J. Yick, B. Mukherjee,D. , Wireless sensor network survey, Computer Networks, Volume 52, Issue 12, 22 August 2008, pp 2292-2330.

[4] K. Sohrabi, J. Gao,V. Ailawadhi,G.J. Pottie, Protocols for self-organization of a wireless sensor network, Personal Communications, IEEE , vol.7, no.5, pp.16-27.

Xiang Su, Jukka Riekki, Janne Haverinen
Department of Computer Science and Engineering
University of Oulu, Finland

Johanna Nieminen
TeliaSonera, Helsinki, Finland

Jukka K. Nurminen
Department of Computer Science and Engineering
Aalto University, Finland

ONTOLOGY ALIGNMENT FOR INTEROPERABILITY ON THE IoT

The Internet of Things is coming, but it needs a semantic backbone to flourish. Some 50 billion devices are expected to be connected to the Internet by 2020, making interoperability a major concern. Most of these devices will be deployed for industrial and public infrastructure domains, where a need for the emergence of standardized domain models, i.e. ontologies, is well recognised. We believe, however, that in the customer segment of IoT that comprises smart homes, smart offices, connected vehicles, and similar, creation of standard ontologies is much more challenging but also less beneficial. Therefore we investigate how IoT environments can function with the help of ontology alignment solutions that discover the mappings between the concepts from two alternative domain models in an automated fashion.

Our work is motivated by a vision of the Internet of Things where 3rd-party software application development for IoT environments like smart homes is as easy and as popular as the development of applications for smartphones nowadays. One barrier is a big number of various and non-interoperable IoT platforms, and too small a market penetration of each. We aim at a solution, therefore, which enables developing applications that are generic in the sense of being able to communicate with sensors and to control actuators connected to the Internet through different platforms. This is in contrast to the present restriction of always developing an application for a very particular IoT platform.

Figure 1, depicting our prototype system setup, exemplifies this concept. Assume one user has a ThereGate gateway and a Z-wave contact sensor, while another user has a Texas Instruments USB dongle and a ZigBee contact sensor. Each platform defines its own format for queries and its own way of describing door open/close events, including different data structures and names for properties ('DoorOpen': 'true' vs. 'action': 'open'). Yet, both users are able to deploy exactly the same application code from an online IoT App Store and successfully run it. Note that interoperability is not burdened on the application or its execution platform, as is in many other approaches. We assume no particular execution platform and the application can define yet another, its own, data representation format. It is a smart proxy in-between the application and the sensor that manages the interoperation. An additional task of the smart proxy is the discovery of appropriate sensors/actuators within an IoT environment to match the requirements of the application.

The central element of our Semantic Smart Gateway Framework (SSGF) is a smart environment registry that contains semantic descriptions of 'things' (a door), connector devices (a contact sensor) and their associations to 'things' (the contact sensor attached to the door), as well as the deployed applications. These descriptions are based on an IoT ontology we developed, which is the extension of W3C Semantic Sensor Network (SSN) ontology, which is, in turn, based on DUL (DOLCE Ultra Light) upper ontology. The IoT ontology only provides vocabulary related to generic sensing/actuating, while for any domain-specific concepts some custom classes are to be used and can be freely defined. For match-making of sensors/actuators and applications, an application's requirements are expressed as SPARQL query patterns. Similarly to semantic descriptions of things and devices, these patterns are defined using our IoT ontology plus some custom classes for domain-specific concepts. An ontology alignment solution is then applied to find the mappings between the custom classes used in device descriptions and application patterns.

The same ontology alignment solution is also utilized for the second, a more complex, alignment task that is the automated transformation of data formats used by an application and a sensor/actuator. Figure 2 depicts the related workflow.

The semantic descriptions of both the device and the application have to include examples for all relevant query and response messages, which can be e.g. XML, JSON, or complex URIs. From an analysis of these example messages, OWL ontology models for the device and the application are generated and heuristically refined. These ontology models are then fed into the ontology alignment solution to discover the concept mappings. Finally, these mappings are used in the run-time by the message translator component of the smart proxy to provide two-way communication message transformation between the application and the devices.

SSGF facilitates automated deployment of generic and legacy IoT software in environments where heterogeneous devices also have been deployed. SSGF functionality can be implemented by an IoT platform provider to enable their platform to run applications not originally designed for it, i.e. to extend the range of applications available to their customers. Alternatively, SSGF can be delivered by an independent party as a service, resulting in a novel "interoperability-as-a-service" paradigm. Practically, this means operating a scalable web portal where the end-users can register their things and devices, as well as deploy application descriptions from app stores. The data traffic between applications and end-users' devices will also go through this web portal.

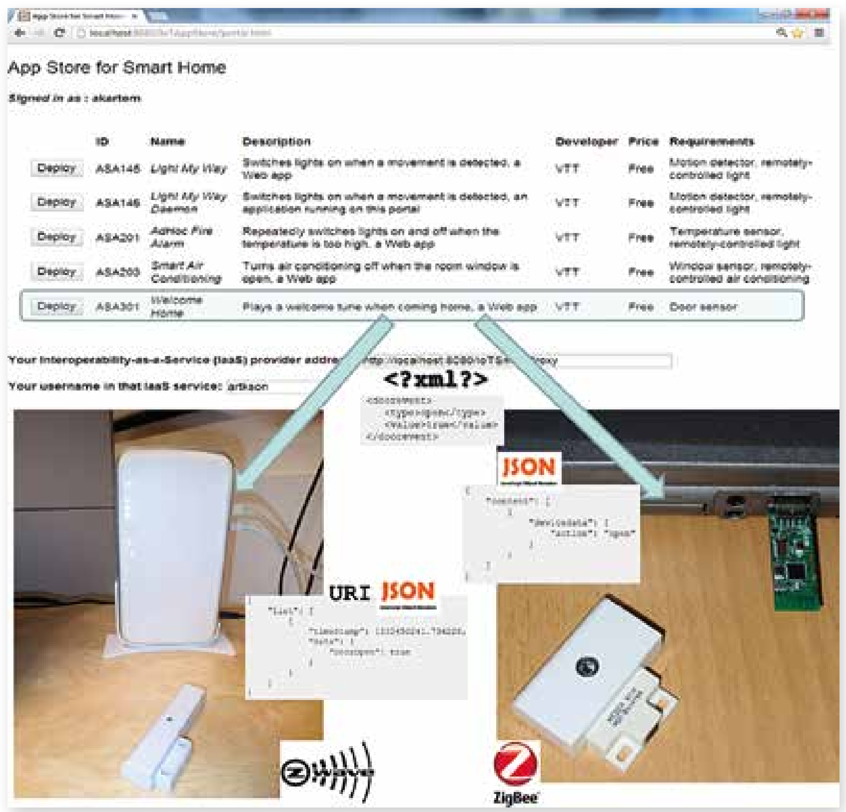

Figure 1. *Deploying an application to two IoT platforms*

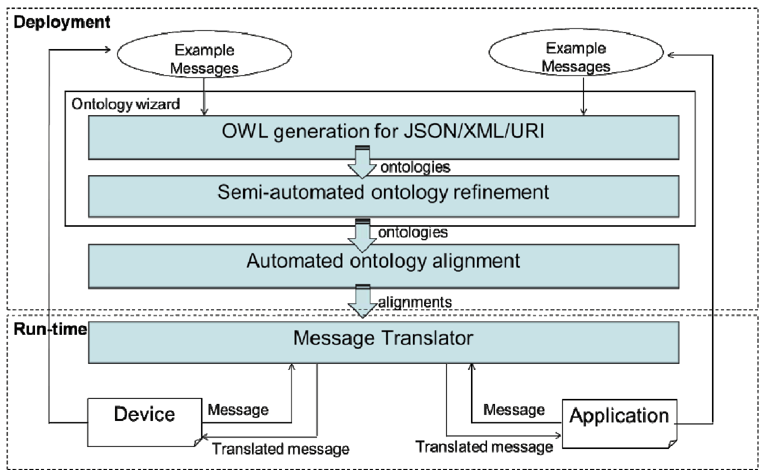

Figure 2. *Workflow for automated translation between data formats*

More technical details about SSGF can be found in [1].
[1] Kotis K. and Katasonov A. (2013) Semantic Interoperability on the Internet of Things: The Semantic Smart Gateway Framework, Int. J. Distributed Systems and Technologies, IGI Global, in press

Artem Katasonov and Konstantinos Kotis,
VTT Technical Research Centre of Finland