5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014)

Connecting IoT Sensors to Knowledge-Based Systems by Transforming SenML to RDF

Xiang Su[a,*], Hao Zhang[b], Jukka Riekki[a], Ari Keränen[c], Jukka K. Nurminen[d], Libin Du[b]

[a]*Department of Computer Science and Engineering, University of Oulu, Finland*
[b]*Institute of Oceanographic Instrumentation, Shandong Academy of Sciences, China*
[c]*Ericsson Research, Finland*
[d]*Department of Computer Science and Engineering, Aalto University, Finland*

Abstract

Applying Semantic Web technologies to Internet of Things (IoT) enables smart applications and services in a variety of domains. However, the gap between semantic representations and data formats used in IoT devices introduces a challenge for utilizing semantics in IoT. Sensor Markup Language (SenML) is an emerging solution for representing device parameters and measurements. SenML is replacing proprietary data formats and is being accepted by more and more vendors. In this paper, we suggest a solution to transform SenML data into a standardized semantic model, Resource Description Framework (RDF). Such a transformation facilitates intelligent functions in IoT, including reasoning over sensor data and semantic interoperability among devices. We present a fishery IoT system to illustrate the usability of this approach and compare the resource consumptions of SenML against other alternatives.

© 2014 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and Peer-review under responsibility of the Program Chairs.

Keywords: Media Types for Sensor Markup Language; RDF; Inference.

1. Introduction

In the Internet of Things (IoT), varieties of things (i.e. objects) around us have Internet addresses and interact with each other to achieve common goals. These objects should be able to serve multiple applications, rather than a single dedicated application. In this paper, we focus on interoperability at the data and knowledge level. Semantic Web technologies can provide machine interpretable meanings for IoT data. Hence, meaning of data can be comprehended unambiguously and additional knowledge can be derived. For example, Resource Description Framework (RDF)[1] is one of the basic knowledge models of Semantic Web and it directly supports advanced models and reasoning techniques. However, these technologies require a considerable amount of computing and communication resources, which are not always available for IoT devices.

---

* Corresponding author. Tel.: +358-29-448-2998 ; fax: +235-8-553-2612.
  E-mail address: Xiang.Su@ee.oulu.fi

Sensor Markup Language (SenML)[2] is an emerging standard for representing sensor measurements and device parameters. As an industry-driven representation, SenML is taking a more and more important role in IoT domains and applications. It is not a proprietary data format; hence, it enables good interoperability among IoT devices from different vendors. Moreover, SenML supports compact formats, i.e. JavaScript Object Notation (JSON)[3] and Efficient XML Interchange (EXI)[4] format for the tiniest devices. JSON might be the most widely used syntax for SenML. When devices have limited communication resources, EXI can be utilized.

In this paper, we tackle the challenge of bridging the gap between semantic representations and SenML. We present our work towards transforming SenML data into the standardized semantic model, RDF. Transforming SenML into RDF would facilitate intelligent IoT applications. For example, data from physical and logical sensors could be analyzed and deduced into actionable knowledge. This would give better understanding about our physical world to human beings and enable creating more value-adding products and services.

Our main contribution are: 1) an approach to convert basic SenML data to RDF. With this approach, current SenML-enabled devices can take the benefit provided by knowledge-based systems without any extra complexity. SenML-enabled devices can be utilized as such and do not need any extra software library or processors. 2) A fishery IoT system, including temperature and salinity sensors and a knowledge-based component, illustrates the usefulness of our approach. With this system, we evaluate SenML against other formats in resource usage aspect, including computing, communication, and energy consumptions.

Most data formats utilized for embedded devices, such as JSON, YAML, comma-separated values, and different binary formats only define data structures and methods for encoding and decoding these data structures. Hence, they cannot be transformed into any knowledge representation in a straightforward manner. On the other hand, some representations produced by the Semantic Web community are potential candidates in the IoT area. For example, Notation 3 (N3)[5], Turtle[6], and N-Triples[7] have good semantic expressive power and are easy to be interpreted. Entity Notation[8] is designed for embedded systems and enables a transformation into Semantic Web models. JavaScript Object Notation for Linked Data (JSON-LD)[9] is an emerging upgrade for JSON. JSON-LD can be utilized as RDF syntax and actually has slightly better expressive power than RDF. However, none of these representations have been designed for embedded devices and constrained application protocols like SenML has. Finally, Constrained RESTful (CoRE) Link Format[10] provides a way to describe resources and attributes of the resources and relationships between links. However these relationships are not suitable as such for performing reasoning for the data. We are not aware of any related work towards transforming SenML into any knowledge representation.

We have reported earlier idea about transforming SenML into semantic representations[11], but here we describe for the first time the details of this approach. We will continue this article with introducing our solution of transforming SenML data into RDF in Section 2. In Section 3, we present a use case about reasoning over SenML sensor data in a fishery system to illustrate the usefulness of our approach and evaluate SenML against other formats. We conclude the paper and suggest future work in Section 4.

2. Transforming SenML Data to RDF

SenML enables connecting IoT devices to the Internet at the data exchange level. SenML is designed for resource-constrained devices, so complex information, such as semantics, has been intentionally left out. A SenML description carries a single base object consisting of attributes and an array of entries. Each entry, in turn, consists of the name of the sensor parameter and attributes such as the time of the measurement and the current value. SenML format can be extended with custom attributes. For example, the Resource Type (rt) attribute can be used to define the type of a resource. This feature makes it possible to include semantic information, while keeping SenML description simple.

The basic structure of RDF statement is *(Subject, Predicate, Object)*. This triple represents a statement of a relationship between the things denoted by the nodes that it links[12]. RDF utilizes URI references and literals as identifiers for representing elements. RDF supports containers and collections to represent complex data structures, and they are ideal solutions for multiple sensor measurements. RDF triples can be used in a straightforward fashion by a knowledge-based system. This makes inference possible over real world sensor measurements.

The core of our approach is enabling a mapping between SenML elements to the RDF model, that is, to a labelled, directed graph. Our design considerations are: firstly, SenML does not Utilize Resource Identifiers (URIs) to the same extent as RDF. URIs are a fundamental building block of RDF; every non-literal data item has its own URI.

Hence, we need a universal identification mechanism for SenML elements. Secondly, one SenML description can be transformed into one or more RDF triples. Thirdly, a namespace needs to be defined in IoT applications. Sensors utilize this namespace and the IoT systems processing sensor data should understand this namespace.

Transforming SenML into RDF requires a unique identification mechanism. SenML elements should be transformed into URIs, and sensor measurements to XML schema data type literals. It is common that all corresponding URIs of SenML elements in one sensor data packet are defined in one name space. Table 1 presents the mechanism to utilize URIs for assigning unambiguous identifiers to SenML elements. This table shows the mapping from SenML elements, their shorthands in JSON, to their corresponding types when transformed into RDF. Details of the meaning of SenML elements can be found from SenML specification[2]. Base Name is usually utilized to identify devices; hence, it is an obvious choice for the Subject of the RDF statement. When a MAC address is utilized in Base Name, it must be transformed into a URI format. For example, a Base Name in URI format can be accessed by linking a prefix with a URN MAC address. Measurement or Parameters element in JSON includes one or more entries of sensor measurements or configuration parameters. Each of these entries can be transformed to one RDF triple. For example, a Base Name can be transformed into RDF Subject, a measurement Name to RDF Predicate, and measurement value to Object. Therefore, the Measurement or Parameters element do not map to any type in RDF (shown in 5th row in Table 1). Resource Type (shown in the last row of Table 1) takes an important role in transforming SenML descriptions into RDF triples. It indicates the type of devices generating the data, that is, the type of the Subject in an RDF statement and will hence be mapped to rdf:type. When ontology reasoning is applied to sensor data, rdf:type will be connected to a class name of an ontology. We define Resource Type as a mandatory element when SenML descriptions are transformed into RDF. This means that every description should include it, unless the receiving peer already knows this information beforehand. When a knowledge-based system already has the knowledge of this sensor and its type information, i.e. this sensor is registered as an individual of certain Class in an ontology, Resource Type element can be left out.

Table 1. Transforming SenML Elements to RDF.

SenML Elements	JSON Shorthands	Types in RDF
Base Name	bn	URI (Subject)
Base Time	bt	xsd:dateTime
Base Units	bu	xsd:int
Version	ver	xsd:int
Measurement or Parameters	e	– (RDF Triples)
Name	n	URI
Units	u	xsd:string
Value	v	xsd:float
String Value	sv	xsd:string
Boolean Value	bv	xsd:boolean
Value Sum	s	xsd:float
Time	t	xsd:dateTime
Update Time	ut	xsd:dateTime
Resource Type	rt	URI (rdf:type)

The transformation allows connecting SenML-enabled IoT sensors to knowledge-based systems with minimal code changes. Sensors can utilize SenML without any additional computation to prepare the data and a simple parsing component can be employed at a knowledge-based system for transforming SenML into RDF. This component can be physically deployed in a gateway or a server machine of IoT systems. An algorithm for implementing such a component has the following steps.

```
STEP 1. Transform SenML elements into their corresponding unique identified elements, normally URIs
and literals. If any prefix or basename is defined in SenML document, it should be concatenated
with element names. This ensures that resources, properties, types, and values are given their
full representations.
STEP 2. Reorganize SenML document into an array of RDF triples. Introduce RDF Containers, RDF
Collections, etc., when needed.
STEP 3. Serialize RDF triples to representations, for example, XML, N3 and JSON-LD. Define XML name
```

```
{"e":[
      { "n": "temperature", "u": "Cel", "v": 22},
      { "n": "salinity", "v": 0.031 }],
  "bn": "http://iot.fi/o#temsalSensor011/"
  "rt": "MarineSensingNode"
  "pr": "http://iot.fi/o#"
}
```

Fig. 1. An Example of SenML Data Produced by a Marine Sensor

```
<rdf:RDF xml:base="http://iot.fi/o"
 xmlns:i="http://iot.fi/o#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:iotterms="http://iot.fi/terms/">
  <i:MarineSensingNode rdf:ID="temsalSensor011">
   <i:temperature>22</i:temperature>
       <iotterms:units rdf:resource = "http://iot.fi/units/Celsius"/>
  <i:salinity>3.1%</i:salinity>
  </i:MarineSensingNode>
</rdf:RDF>
```

Fig. 2. Corresponding RDF/XML representation the SenML data shown in Figure 1
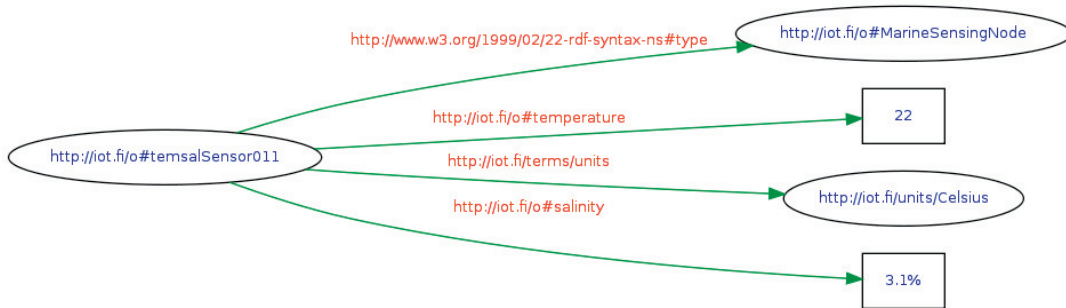

Fig. 3. RDF Graph of the Marine Sensor Data Example

spaces, and shorthands, if needed.

This algorithm is suitable for SenML in JSON and XML format. When SenML has EXI format, one more step is needed to convert data from EXI to XML. Some existing libraries can be utilized to perform this conversion in a straightforward manner.

Figure 1 presents an example of SenML data produced by a tiny marine sensor used in our fishery IoT application. It shows temperature and salinity in the local environment around this sensor. This device has the device ID temsalSensor011 (with "bn") and the resource type MarineSensingNode (with "rt"). For solving any potential conflict in global level IoT systems, one prefix element ("pr": "http://iot.fi/o#") is defined for the namespace of this sensor. Similar to Resource Type, a prefix does not need to be transferred in every SenML description, if this type of optimization can be agreed between the IoT sensors and the IoT applications using the data.

Figure 2 presents the RDF/XML representation that this SenML data can be transformed into. It should be noted that measurement values are all in XML Schema data types, and units of measurement values are defined separately. Figure 3 shows the corresponding RDF graph of the RDF/XML description presented in Figure 2.

It is common that one SenML description can be transformed into several RDF statements. We do not study how to transform RDF statements into SenML in this paper, though action information can be sent to IoT devices in such a way. RDF has much stronger expressive ability than SenML; hence, it is too complex for many resource-constrained IoT devices to understand action information when the full power of RDF is considered.
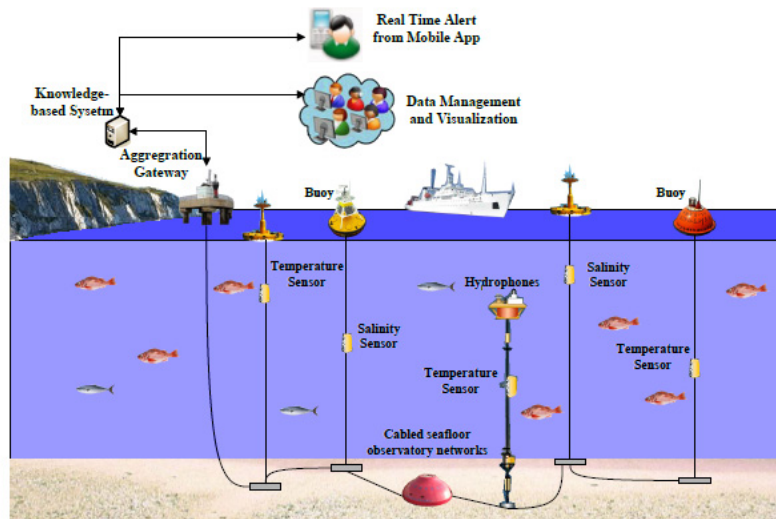
Fig. 4. A Typical System in Fishery scenarios.

3. Implementation and Analysis

IoT technologies enable smart applications and services in a variety of domains. Recently, utilization of IoT technologies in the marine and fishery domain is attracting attention from academia and industry. One typical scenario is monitoring water quality of fish farms. Sensors can observe variables such as temperature, pH value, salinity, and dissolved oxygen concentration. Knowledge models can be applied to analyse sensor data and warn stakeholders in the case of anomalous measurements. Zhou et al. introduce a wireless system for water quality control[13]. Crowley et al. introduce a web-based real-time temperature monitoring of shellfish catches with a wireless sensor network[14]. SEMAT project[15] is a multidisciplinary program developing wireless sensor networks to collect, store, process, and interpret data in coastal systems. An early warning system of tsunami[16] utilize sensor networks and semantic processing techniques. However, although semantic tools, such as Jena, and formats, such as RDF and SensorML, are used in some of these projects, SenML is not utilized.

Our approach is utilized in fish industry for monitoring ambient coastal conditions for pagrus and making decision based on the situation at hand. Temperature and salinity can affect feeding efficiency of larvae of pagrus by influencing processes such as metabolism, oxygen consumption, behavior, swimming speed, and gut evacuation time[17]. To simplify our study, temperature and salinity are considered to be independent of each other in their effect on larval survival, growth and swim bladder inflation.

Figure 4 shows a typical system deployment in fishery scenario. Marine sensors and other devices are deployed for monitoring the temperature and salinity of ambient coastal conditions. These devices are connected with cabled seafloor observatory networks and send measurements to an aggregation gateway in SenML format. This gateway fuses the data and forwards it to a knowledge-based system. The knowledge-based system transforms SenML measurements into RDF statements (with the algorithm presented in Section 2) and integrates these statements into a domain ontology for reasoning. Real time temperature alert or salinity alert will be sent to a client application of the user's mobile phone, and then they can employ instruments when necessary. Moreover, data management and visualization tools are utilized to store and visualize the data.

In the reasoning component of the knowledge-based system, we utilize Semantic Web Rule Language (SWRL)[19] rules for deducing alerts and reminders based on IoT sensor measurements. Alert is the emergency information that needs to be handled immediately, and reminder is warning information that should be paid attention to. Alerts and reminders will be shown together with location of sensors. Figure 5 presents a set of SWRL rules for different temperature and salinity values. Temperature related rules show that alerts are sent when local temperature of fish

```
Temperature Rule 1:
Implies(Antecedent temperature(?sensor, ?temp) location(?sensor, ?tank) swrlb:greaterThan(?temp, 27)
        Consequent(sendAlert( "Alert: check temperature of the water in ?tank")))

Temperature Rule 2:
Implies(Antecedent temperature(?sensor, ?temp) location(?sensor, ?tank) swrlb:lessThan(?temp, 18)
        Consequent(sendAlert( "Alert: check temperature of the water in ?tank")))

Salinity Rule 1:
Implies(Antecedent salinity(?sensor, ?sal) location(?sensor, ?tank) swrlb:greaterThan(?sal, 0.033)
        Consequent(sendAlert( "Alert: check salinity of the water in ?tank")))

Salinity Rule 2:
Implies(Antecedent salinity(?sensor, ?sal) location(?sensor, ?tank) swrlb:lessThan(?sal, 0.008)
        Consequent(sendAlert( "Alert: check salinity of the water in ?tank")))

Salinity Rule 3:
Implies(Antecedent salinity(?sensor, ?sal) location(?sensor, ?tank)
        swrlb:lessThanOrEqual(?sal, 0.016) swrlb:greaterThanOrEqual(?sal, 0.008)
        Consequent(sendReminder( "Reminder: check salinity of the water in ?tank")))
```

Fig. 5. A Set of Rules for Temperature and Salinity

tank is higher than 27 degrees or lower than 18 degrees. Salinity related rules show that alters are sent when local salinity is higher than 33‰ or lower than 8‰. Moreover, reminder information is sent when salinity is between 8‰ and 16‰. Although this example is simple compared with complex real ambient coastal conditions, it nevertheless illustrates the usefulness of our approach.

With a marine sensor, we evaluate resource usage of SenML against other alternatives, including RDF/XML, N3, N-Triples, and JSON-LD. This sensor node consists of an MSP430F2481 microcontroller with 4KB RAM, a thermometer, a salt meter, and a RS-232 transceiver. All other messages were created by filling the data values in a string, but SenML/EXI messages were encoded from SenML/XML using schema-less mode of the "Embeddable EXI implementation in C" software[18]. Sensor data is delivered to an aggregation gateway via a cabled seafloor network. Sensor data decoding is done on a resource rich server, so resource consumptions of decoding, including transforming SenML into RDF, is not considered. In our experiment, the sensor send 100, 500, and 1,000 messages with different formats. We compare message lengths, cycles of microcontroller, and energy consumptions of microcontroller and transceiver with these messages.

In Figure 6(a), we compare overall lengths of 100 messages about temperature and salinity sensor data in the fishery system. Communication resources needed for sending these messages scale linearly with the lengths of messages. RDF/XML, N3, and N-Triples are semantic languages designed for Web applications, they are clearly longer than SenML and JSON-LD. JSON-LD is a lightweight alternative, and still can be utilized as a serialization of RDF model. Our example shows the packet lengths of SenML/JSON contains about 42% of the characters that the corresponding RDF/XML representation, while keeping the same semantics. SenML in EXI can be even more compact, which is only about 28% of the characters of corresponding RDF/XML packets. Figure 6(b) presents the amount of CPU cycles needed to generate one message by this sensor. It is noticed that generating SenML/EXI messages requires more cycles than other alternatives.

Figure 7 presents a comparison of energy consumptions of this sensor. We calculate energy consumption of at MSP430F2481 microcontroller at standby mode and RS-232 transceiver at active mode. These are normal modes when this sensor actively sends data. The energy consumption of microcontroller is strongly dependent on MCU cycles and transceiver on message lengths. The transceiver works with the data rate of 9600 bps. Compared with other alternatives, generating SenML/JSON requires least microcontroller energy and overall energy. When communication resources are rather limited, SenML/EXI is an ideal candidate, because it requires least communication energy. Different formats of SenML are potential to be adopted when sensors are resource-constrained. Therefore, our approach of transforming SenML into RDF keeps SenML simple and resource efficient, while enabling Semantic Web technologies.
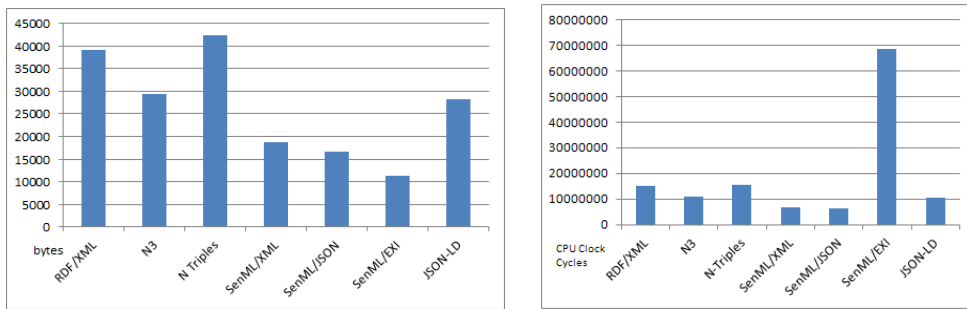
Fig. 6. (a) Comparison of packet lengths of different data formats; (b) Comparison of microcontroller cycle of different data formats.
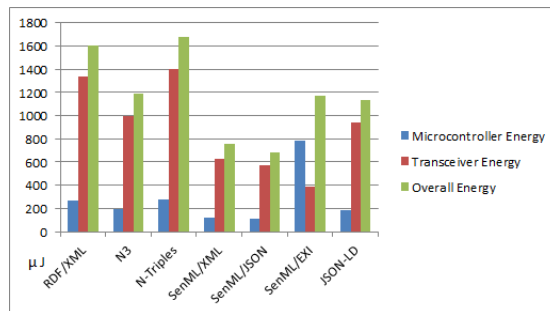

Fig. 7. Comparison of computation and communication energy of different data formats.

4. Discussion

In this paper, we presented an approach for connecting SenML-enabled IoT sensors to semantic knowledge-based systems. We introduced details of this approach, design considerations, a real use case study from fish industry, and a comparison of SenML with other formats. Our use case illustrated the usability of this approach and our evaluation compares SenML against other alternatives and shows how resource-efficiency SenML can be. By connecting a large amount of resource-constrained IoT sensors utilizing SenML to Web and Semantic Web, this approach can enable a large variety of intelligent IoT applications in different domains. Our approach focuses on simplicity when possible. The aim is that IoT applications take the benefit of semantics and Web technologies all the way, even though many objects are resource-constrained.

IoT technologies provide promising approaches for marine and fishery domains. However, special characteristics of these domains require considering also common communication, tracking, localization, and energy harvesting techniques [20]. SenML is a flexible data format that can be used with different networks and architecture, including seafloor observatory networks and the Internet. Connecting SenML-enabled IoT sensors to knowledge-based systems bridges the gap from sensors to decision making applications at the data exchange level.

Our approach provides a natural way to integrate deployed systems that already use SenML into knowledge-based systems. No extra software library or processors are needed to utilize this approach, only a resource type property of devices need to be specified clearly on a gateway or a server machine. This approach offers a starting point to link existing SenML-enabled networked sensors to enable novel intelligent applications. IoT sensors can simply send SenML packets without any knowledge of RDF, but still a knowledge-based system can understand and utilize these packets.

One of the most important considerations to enable scalability is the utilization of unique identifiers. Namespaces are defined in IoT applications; hence, every SenML element and measurement data can be represented unambiguously. However, one challenging task will be handling emerging semantics from different IoT applications. Emergent semantics enables an approach to construct knowledge base in the bottom-up way, in which semantically related peers are discovered and linked together during the normal operation of the system. For example, sensor data used by a

fishery IoT application and sensor data used by a tsunami warning system can be from the same marine area. It is important to know what additional knowledge can be produced when such semantic sensor data is combined.

We focus on transforming from sensor measurements in SenML format into RDF. When comparing SenML with RDF in semantic level, RDF has richer expressive power that cannot be supported by SenML. Some complex structure supported by RDF cannot be converted to SenML. Hence, it is a challenge for actuators in an IoT system to understand action information, when full power of RDF is utilized. Another challenge is to study an approach to determine datatypes with values in certain context. For example, when a gateway receives information that temperature is 0 from a seafloor marine sensor, it can understand the fact that the temperature measurement is in Celsius, rather than in Fahrenheit.

Resource usage can be decreased by optimizing communication and system architecture. For example, when a sensor and a knowledge-based system agree a namespace, prefixes (e.g. "pr": "http://iot.fi/o" in the SenML example shown in Section 2) do not need to be transferred in every SenML description. Similarly, when a knowledge-based system knows that temsalSensor011 has resource type BuoySensingNode, this element does not need to be sent. We reported our early experiments about resource usage of several well-designed data formats[21]. In the future, we will measure the overall energy consumption of our approach in real world deployments and consider its limitations.

Acknowledgment

References

1. F. Manola, E. Miller, RDF Primer W3C Recommendation, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.
2. C. Jennings, Z. Shelby, J. Arkko, Media Types for Sensor Markup Language (SENML), http://tools.ietf.org/id/draft-jennings-senml-10.txt. (Work in progress, Last updated: 2013-04-25).
3. D. Crockford, The application/json Media Type for JavaScript Object Notation (JSON), http://www.ietf.org/rfc/rfc4627.txt.
4. World Wide Web Consortium, Efficient XML Interchange (EXI) Format 1.0, http://www.w3.org/TR/exi/.
5. T. Berners-Lee, Notation 3 Logic, http://www.w3.org/DesignIssues/Notation3.html/.
6. D. Beckett, T. Berners-Lee, E. Prud'hommeaux, and G. Carothers, Turtle Terse RDF Triple Language, http://www.w3.org/TR/2012/WD-turtle-20120710/.
7. World Wide Web Consortium, N-Triples, W3C RDF Core WG Internal Working Draft, http://www.w3.org/2001/sw/RDFCore/ntriples/.
8. X. Su, J. Riekki, J. Haverinen, Entity Notation: Enabling Knowledge Representations for Resource-Constrained Sensors, Personal and Ubiquitous Computing, 16(7)(2012) 819-834.
9. M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, N. Lindström, JSON-LD 1.0 A JSON-based Serialization for Linked Data, http://www.w3.org/TR/json-ld/.
10. Z. Shelby, Constrained RESTful Environments (CoRE) Link Format, https://datatracker.ietf.org/doc/rfc6690/. (Last updated: 2013-10-17)
11. X. Su, J. Riekki, A. Keränen, J.K. Nurminen, Transforming SenML Sensor Data to Semantic Representations. IoT Magazine Finland. *Accepted*.
12. World Wide Web Consortium, Resource Description Framework (RDF): Concepts and Abstract Syntax, http://www.w3.org/TR/rdf-concepts/.
13. X. Zhu, D. Li, D He, J. Wang, D. Ma, F. Li, A Remote Wireless System for Water Quality Online Monitoring in Intensive Fish Culture, Elsevier Computers and Electronics in Agriculture 71 (Supplement 1) S3-S9.
14. K. Crowley, J. Frisby, S. Murphy, M. Roantree, D. Diamond, Web-based Real-time Temperature Monitoring of Shellfish Catches using a Wireless Sensor Network, Sensors and Actuators A: Physical 122(2) (2005) 222-230.
15. T. Myers, I. Atkinson, R. Johnstone, Semantically Enabling the SEMAT Project: Extending Marine Sensor Networks for Decision Support and Hypothesis Testing, in: Proc. of CISIS-2010, 2010, pp 974-979.
16. K. Ramar, T.T. Mirnalinee, An ontological representation for Tsunami early warning system, in: Proc. of ICAESM-2012, pp 93-98.
17. D. Stewart Fielder, W. J. Bardsleya, G. L. Allana, P. M. Pankhurstc The effects of salinity and temperature on growth and survival of Australian snapper, Pagrus auratus larvae, Aquaculture 250(1-2) (2005) 201-214.
18. Embeddable EXI Processor in C, http://exip.sourceforge.net/.
19. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/SWRL/.
20. M. C. Domingo, An overview of the internet of underwater things, Journal of Network and Computer Applications 35(6) (2012) 1879-1890.
21. X. Su, J. Riekki, J. K. Nurminen, J. Nieminen, M. Koskimies, Adding Semantics to Internet of Things. Concurrency and Computation Practice and Experience, DOI: 10.1002/cpe.3203.