Lowering knowledge : Making constrained devices semantically interoperable

Nicolas Seydoux[1,2,3], Khalil Drira[2,3], Nathalie Hernandez[1], Thierry Monteil[2,3]

[1] IRIT Maison de la Recherche, Univ. Toulouse Jean Jaurès,
5 allées Antonio Machado, F-31000 Toulouse
{nseydoux,hernande}@irit.fr
[2] CNRS, LAAS, 7 avenue du Colonel Roche,
F-31400 Toulouse, France
{nseydoux,khalil,monteil}@laas.fr
[3] Univ de Toulouse, INSA, LAAS, F-31400, Toulouse, France

Abstract. Semantic interoperability is an issue in heterogeneous IoT systems. The limited processing power and memory storage of constrained IoT nodes prevents them from handling enriched data. This paper proposes a method to lower complex knowledge representations into simpler structured data, based on the reuse of lifting mappings from data schemas to semantic models.

1 Interoperability in the IoT

The Internet of Things (IoT) is in rapid expansion and deploying at a global scale: projections predict up to 50 billion devices connected in the next ten years [3]. Domains impacted by the IoT are very diverse: agriculture, domotics, e-health, smart cities and manufacturing... The diversity of these domains implies a diversity of data models, leading to a lack of semantic interoperability. To overcome the gap between ad-hoc data models, shared machine-understandable knowledge representation is needed. The wide adoption of ontologies such as W3C's SSN[4] and the emergence of initiatives such as LOV4IoT, introduced in [4], show the potential of semantic web technologies for the IoT.

However, knowledge expressed in the W3C formalisms (RDF, OWL...) are heavier to process and exchange than ad-hoc models. This is antithetic with the deployment of very constrained devices that have a low processing power and a restricted bandwidth. The most constrained nodes of the IoT are sensors, which convert physical world signals into virtual measures, and actuators, which transform virtual commands into physical world effects (light, temperature, motor...). These nodes cannot produce or consume enriched data, so making the system semantically enabled requires a transformation from data to knowledge (data enrichment, for sensor observations) and from knowledge to data (knowledge lowering, for actuator commands).

In the remainder of this paper, after introducing **mapping-based data enrichment** methods, we propose a lowering method to capture rich knowledge into simpler data formats so that they can be processed by constrained devices.

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

This method is based on the **reuse of existing mappings** used for generating enriched data instead of dedicated ones, thereby reducing the cost of lowering. A qualitative evaluation of our approach is then provided.

2 Enrichment and lowering: data management in the IoT

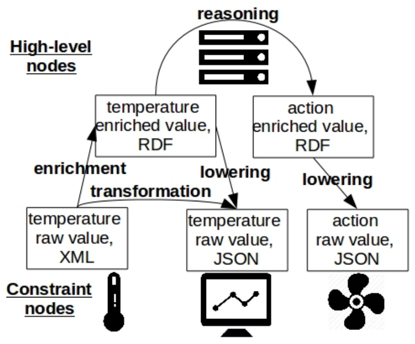
reasoning
High-level nodes
temperature enriched value, RDF
action enriched value, RDF
enrichment
lowering
lowering
transformation
temperature raw value, XML
temperature raw value, JSON
action raw value, JSON
Constraint nodes

Fig. 1. Illustration of data enrichment and lowering

Figure 1 gives an overview of data management issues in the IoT by a use case: data measured by a temperature sensor is displayed by an application using a different data format, and it is also used to control a fan. By definition, IoT nodes produce and consume structured data (raw data value on fig. 1). [7] and [2] exploit the **implicit semantics** encoded in the **syntactic data model** by aligning it to ontologies using semantic annotations. Data lifting (**enrichment** on fig. 1) can thus be done by mapping the schema to an ontology rather than the data itself. To reduce the payload when transferring data, the schema can be accessed and manipulated only when necessary. On the other hand, existing approaches for data **lowering** (from a semantic representation to a syntactic one) rely on dedicated mappings, as in [5] or [1]. To bring interoperability to lower nodes, [6] proposes an ontology-driven **data transformation** approach, i.e. the presentation of a piece of data in a different target format (also illustrated on fig. 1). We propose to use schema mappings to ontologies initially meant for data enrichment for knowledge lowering.

Shared schemas are syntactic interoperability providers. Many standards, especially for the IoT, normalize schemas: oneM2M[5]with the SDT (Smart Device Template)AllSeen-AllJoyn[6], or OCF[7] with OIC [8]. Therefore, using schema mappings for enrichment or lowering is suitable for an IoT system, as it limits bandwidth consumption when exchanging data, integrates constrained devices, and brings semantic interoperability. However, manually aligning a schema to an ontology is a **time consuming process**, and it has to be done for both enrichment and lowering. Our hypothesis is that if a schema only aligned for upstream

[5] http://onem2m.org/
[6] https://allseenalliance.org/
[7] http://openconnectivity.org
[8] http://openconnectivity.org

enrichment meets a set of requirements, the mappings can be used for knowledge lowering as well, cutting by half the work required to make constrained devices semantically enabled.

3 A pivot-tree based approach to mapping reversal-based knowledge lowering

Our approach to schema alignment reversing is divided into two steps: transformation of an RDF graph into an abstract pivot tree, and serialization of this pivot tree into a concrete syntax. The pivot tree is a **language-agnostic** representation of the final tree (either XML, JSON or an URI for REST interfaces) with no concrete syntax. The pivot tree associates the knowledge expressed in the RDF graph (with its explicit semantics) to the tree-like structure of the targeted schema, so that its semantics is no longer implicit.Algorithm 1 shows the creation process of the abstract tree from the annotated schema and the target ontology.

Algorithm 1 Abstract tree creation

```
 1: procedure buildAbstractTree(Schema $s, Graph $g, Resource $individual)
    match is a function that finds the schema element corresponding to the root individual
 2:     $schemaRoot ← match($s, $individual)
 3: build constructs an abstract node from an rdf node and a matching schema node
 4:     $abstractRoot ← build($schemaRoot, $individual)
 5:     for $property in $graph.triples($individual, $property, $object) do
 6:         if $schemaRoot.hasAnnotatedDescendant($property) then
 7:             if $property is ObjectProperty then
 8:                 $abstractNode ← buildAbstractTree($schemaRoot, $g, $object)
 9:                 $abstractRoot.buildObjProp($s, $g, $property, $abstractNode)
10:             if $property is DataProperty then
11:                 $abstractRoot.buildDataProp($s, $g, $property, $object)
```

This approach makes assumptions: the knowledge and the schema must meet some requirements. The target format must be a tree-like structure aligned to semantic resources, schema root(s) must be mapped to classes and meaningful inclusions in the schema must be mapped to relationships. There should be no ambiguity in the mapping: when transforming the RDF graph into the abstract tree, every relation from a node in the graph should be associated to one relation from the corresponding node in the schema.

4 Preliminary results

To evaluate our solution, we measured the compliance of existing schemas with our requirements, and conducted a qualitative comparison with existing works (cf. fig. 1). Different sources of data schemes were used to assess the hypothesis for a reversable mapping : schemes provided as part of standards specifications (AllJoyn), and a sample of schemes from an online API repository[9]. Overall,

[9] http://www.programmableweb.com

these schemes met our requirements, or needed minimum backward-compatible transformation (such as the semantic annotation for enrichment).

Table 1. Qualitative comparison of approaches

	Supported scenario	**Code base**	**Execution time**	**Generation required**	**Supports inference**	**Flexible**	**Schema type applicability**	**Mapping required**
Direct schema transformation	Transformation	Small	Fast	No	No	No	Any to any	None
Schema transformation generation [6]	Transformation	Large	Fast	Yes	No	Partially	Annotated schema	Two-way
Two-way mapping [1]	Transformation and lowering	Large	Slow	No	Yes	Yes	Annotated schema	Two-way
Mapping reversal (our approach)	Transformation and lowering	Large	Slow	No	Yes	Yes	Restricted annotated schema	One way

Despite their runtime efficiency, transformation-only methods are not suitable because raw data is not enriched and cannot be reasoned upon. Our approach covers less schema than the two-way mapping approach because of the hypothesis that must be validated by the schema, but it is suitable for the schema at stake in the IoT as they are standardized, and requires less mapping work than any other approach.

5 Conclusion and perspectives

To bring semantic interoperability to constrained devices, data enrichment and knowledge lowering are required. We propose in this paper a knowledge lowering technique that reuses schema mappings dedicated to data enrichment in order to reduce the cost of integrating constrained devices in an IoT network. Future work will include a more complete evaluation, as well as an integration of this approach to an autonomic agent for the control of a connected apartment.

References

1. Bischof, S., Decker, S., Krennwallner, T., Lopes, N., Polleres, A.: Mapping between RDF and XML with XSPARQL. Journal on Data Semantics 1(3), 147–185 (2012)
2. Ferdinand, M., Zirpins, C., Trastour, D.: Lifting XML Schema to OWL. Web Engineering (3140), 354–358 (2004)
3. Ganz, F., Puschmann, D., Barnaghi, P., Carrez, F.: A Practical Evaluation of Information Processing and Abstraction Techniques for the Internet of Things. IEEE Internet of Things Journal 2(4), 340–354 (2015)
4. Gyrard, A., Serrano, M., Atemezing, G.A.: Semantic web methodologies, best practices and ontology engineering applied to Internet of Things. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). pp. 412–417. IEEE (2015)
5. Kopecký, J., Vitvar, T., Bournez, C., Farrell, J.: SAWSDL: Semantic Annotations for WSDL and XML Schema. IEEE Internet Computing 11(6), 60–67 (2007)
6. Köpke, J., Eder, J.: Semantic annotation of XML-schema for document transformations. On the Move to Meaningful Internet Systems: OTM 2010 Workshops 6428 LNCS, 219–228 (2010)
7. Sheth, A., Henson, C., Sahoo, S.S.: Semantic Sensor Web. In: IEEE Internet Computing. vol. 12, pp. 78–83 (2008)