Semantic Data Integration on the Web of Things

Victor Charpenay
Siemens AG
Munich, Germany
victor.charpenay@siemens.com

Sebastian Käbisch
Siemens AG
Munich, Germany
sebastian.kaebisch@siemens.com

Harald Kosch
Universität Passau
Passau, Germany
harald.kosch@uni-passau.de

ABSTRACT

This paper presents a method for the integration of data originating from sensors and actuators that follow different formalisms, although they semantically interlap. We tested our approach one three Web of Things standards published respectively by the Open Mobile Alliance (OMA), the Open Connectivity Foundation (OCF) and the oneM2M foundation.

Our method extensively relies on Semantic Web technologies. First, observing that all standards provide a JSON representation of the data they specify, we generate an equivalent RDF representation by exploiting features of the recent JSON-LD format. We then define SPARQL inference rules, part of the new SHACL specification, to align the resulting RDF data with a unified ontology we call the *Web of Things cloud*. This ontology includes concepts from the SOSA, SSN, SAREF and SEAS ontologies.

We evaluated our method by measuring the semantic similarity that exists between the standards OMA, OCF and oneM2M define. Our experiments show that the overlap between these standards is limited. Although all of them focus on the home & building automation domains, the schemas they provide cover different device types.

Author Keywords

Web of Things; Internet of Things; oneM2M; OCF; IPSO; LWM2M; JSON-LD; OWL; SPIN; SHACL; Semantic integration.

INTRODUCTION

The Web of Things (WoT) was first introduced almost a decade ago [18]. Since then, it has become a mature technology field endorsed by the industry. OneM2M[1] and the Open Connectivity Foundation (OCF)[2] are two industrial consortia publishing information models and communication standards that follow the principles of the Web of Things: sensors and actuators implementing these standards shall expose RESTful interfaces on the Web via HTTP and WebSocket (or similar protocols such as CoAP, the Resource Constrained Application Protocol [35]).

Typically, a Web agent could request the switch status of a oneM2M device, exposed as a Web resource, and get the following JSON representation containing `"powerState": false` to indicate that the switch is currently turned off:

```
{
  "containerDefinition":
    "o.o.h.m.binarySwitch",
  "powerState": false,
  "childResources": [...]
}
```

Similarly, the switch could be turned on by sending a command to the same resource with another JSON object in the request payload. An OCF device would also expose data via a Web resource but with a different representation of the switch state, as follows:

```
{
  "rt": ["oic.r.switch.binary"],
  "id": "someUniqueID",
  "value": false
}
```

Here, the comparison highlights what remains a critical issue in the Web of Things, namely interoperability. OneM2M and OCF rely on the same (Web) architecture, the same communication protocols and the same encodings, and yet, a generic Web agent would have to deal with two distinct information models to be able to infer the state of a WoT device. In this particular case, a mapping between the two representations is easy but, in the general case, there is no guarantee that a mapping is possible, for lack of contextual meta-data.

This issue of interoperability has been a concern since the coming of the Internet of Things (IoT) [10, 22, 34]. Interoperability, in a broad sense, is the ability of distinct systems to exchange and use information [1], it is one of the architectural requirements of the IoT [2]. In this paper, we deal more specifically with the subsequent problem of data integration, that is, the transformation between `"powerState": false` and `"value": false`, in this particular example.

Currently, standardization bodies like oneM2M and OCF define their own models without considering formal alignment with each other. In this paper, we review the models of oneM2M, OCF and the Open Mobile Alliance (OMA) — another standardization body with similar objectives— in or-

[1] http://onem2m.org/
[2] http://openconnectivity.org/

IOT '18, October 15–18, 2018, Santa Barbara, CA, USA

ISBN 978-1-4503-6564-2/18/10…$15.00

DOI: https://doi.org/10.1145/3277593.3277609

der to formally align them. More specifically, we align these models to a reference ontological model centered around the Semantic Sensor Network (SSN) ontology.

Section 2 presents the work of oneM2M, OCF and OMA, as well as the state-of-the-art regarding ontologies for the IoT. We then present in Sec. 3 how we used Semantic Web technologies (RDF, SPARQL) to semantically integrate data produced by devices implementing different standards. We evaluate our approach in Sec. 4 by comparing oneM2M, OCF and OMA with each other, both semantically and lexically. We then conclude in Sec. 5.

RELATED WORK

Web Technologies for the IoT

The original idea of the Web of Things is the following: exposing low-power IoT devices on the Web can increase their integration to open systems (via Web APIs) and the composability of the services they provide (e.g. Web service mashups) [17, 37]. Since then, it has been the subject of many scientific workshops, first emerging in the field of pervasive computing and sensor networks [19], and later reaching the Semantic Web community [16]. In an attempt to summarize the outcomes of this research in a collection of Web standards, the W3C launched a working group for WoT in 2016 [26].

As stated in its charter, the WoT framework is not supposed to replace or subsume the work of other standardization bodies. The latter either provide extended models in a specific domain (such as Project Haystack[3] or the Fairhair alliance[4] in the domain of building automation) or provide a generic reference architecture. OneM2M, OCF and OMA fall in the latter category. These three have the peculiarity that the gap between their respective standards is small. As already mentioned, they all rely on the Web architecture by exposing RESTful interfaces but, in addition, they also all have a binding to the same protocol (CoAP) and support the same format (JSON). The only issue they do not virtually address is that of semantic interoperability, ideal for our study. These three standards also have the merit of providing models for the same application domain: home automation. We review these models in the following.

Among the numerous standards published by OMA, the ones that directly relate to WoT are grouped under the name OMA SpecWorks[5]. SpecWorks includes the Lightweight Machine-to-Machine (LWM2M) specification, which specifies an object and resource model that can be specialized by third-parties [7]. OMA SpecWorks itself provides objects for various sensors and digital/analog industrial devices[6]. All models are published on a registry managed by OMA[7]. LWM2M relies on the SenML data model [6], which includes a JSON serialization [23].

[3]https://project-haystack.org/

[4]https://www.fairhair-alliance.org/

[5]https://www.omaspecworks.org/

[6]Initially developed by the IPSO alliance, independent from OMA.

[7]http://www.openmobilealliance.org/wp/OMNA/LwM2M/LwM2MRegistry.html

Table 1. Meta-models and data formats.

	oneM2M	oneIoTa	IPSO
Meta-model	SDT	OAS	LWM2M
Model	HAIM	oneIoTa	IPSO
Data format	XSD	JSON schema	senML

Similarly, OCF publishes shared models on the oneIoTa platform[8]. These models are machine-readable Web API specification templates, following the Open API Specification (OAS) format [8]. The default serialization for OAS is JSON.

Finally, the oneM2M organization publishes several domain-specific models, all based on the Smart Device Template (SDT) framework, specified by the Home Gateway Initiative (HGI)[9]. So far, oneM2M has published SDT models for the domain of home appliances only, in a specification called the Home Appliance Information Model (HAIM) [5]. This specification also includes XML schema definitions (XSD) and a mapping from XML to JSON.

As this review shows, none of these standardization bodies share the same baseline, although they all tend to reuse existing works. Our summary in Table 1 highlights their heterogeneity. However, collaboration across standardization bodies exists, as their members acknowledge the need for interoperability in IoT systems. This statement was the conclusion of the Workshop on IoT Semantic/Hypermedia Interoperabilty (WISHI)[10], an event organized by the Internet Engineering Task Force (IETF) where all organizations cited in this section were represented. The conclusions of this workshop were also the starting point of our present work.

Web of Things Ontologies

As mentioned earlier, the Semantic Web community also deals with the Web of Things, mostly regarding semantic interoperability and integration. While, in the fields of pervasive computing and sensor networks, the outcomes of research are being transferred to the industry via standardization, the question of semantic integration on the Web of Things is still an open research question. So far, Semantic Web technologies have been identified as the most relevant vector of progress towards that objective [10, 22].

Among others, RDFS and OWL are W3C recommendation to express so-called *ontologies*, which correspond to machine-readable concept definitions exposed on the Web. Data integration can be achieved by linking multiple sources to a reference ontological model. The W3C WoT group is working on giving WoT servers a means to expose their capabilities by referencing ontologies via the Thing Description (TD) model [24]. It happens that oneM2M also includes OWL definitions in its standards (the oneM2M base ontology [4]). OCF and OMA, however, do not.

In a preliminary study on candidate ontologies for WoT, we identified a lack of homogeneity in the state-of-the-art, espe-

[8]http://oneiota.org/

[9]http://www.homegatewayinitiative.org/

[10]https://github.com/t2trg/2017-07-wishi

cially regarding the modeling of Web services and physical quantities [11]. However, most ontologies were already providing semantic alignment with the Semantic Sensor Network (SSN) ontology, first incubated by the W3C [12] and now an official W3C recommendation [21]. A lightweight version of SSN is also part of the standard: the Sensor, Observation, Sample, and Actuator (SOSA) ontology.

SSN provides an in-depth model to specify sensor measurements, the features of interest being observed, system deployments and the conditions of data acquisition. Alignment with other vocabularies is also provided. In many places, however, the terms SSN defines are placeholders for future alignments with domain-specific vocabularies. For instance, there is no definition for the concept of temperature in SSN, nor is there a definition of a binary switch (as in our introductory example). It only includes the more abstract concepts of 'property' and 'device'.

A recent workshop on semantic interoperability and standardization in the IoT (SIS-IoT) resulted in several proposals for an alignment of SSN with other vocabularies: SAREF and SEAS [27, 30]. The Smart Appliance Reference (SAREF) ontology defines both concepts of 'temperature' and 'binary switch', as well as other elements of building automation systems. This work was mostly driven by the industry and became a standard by the European Telecommunications Standards Institute (ETSI) [3]. The standard also includes an alignement of the oneM2M base ontology to SAREF. Contributions to the SIS-IoT workshop include an alignment between SAREF and SSN, which states e.g. that 'temperature' is a sub-class of 'property' and 'binary switch' a sub-class of 'device'. Indirectly, SAREF also aligns with the Ontology of units of Measure (OM) by reusing some of its units. OM is not a standard but it is a comprehensive and well-maintained set of axioms.

The Smart Energy Aware System (SEAS) ontology provides both a generalization of SSN and several specializations (e.g. for the building automation, energy and environmental domains) [28]. The ontology was designed by active contributors of SSN and an alignment between the two ontologies is provided. In SIS-IoT, a direct alignment between SEAS and SAREF was also proposed and an official endorsement of SEAS by ETSI is planned. In addition, an alignment between SEAS and the TD vocabulary (which is also defined as an OWL ontology) was proposed.

SSN, SAREF and SEAS are not the only ontological models for the IoT in the state-of-the-art. The Linked Open Vocabulary initiative for the IoT (LOV4IoT) indexes all ontologies on the Web related to the IoT [20]. However, the ontologies listed on LOV4IoT are of varying quality and some of them are not actively maintained. In contrast, all ontologies considered in SIS-IoT are —or will be— standards, i.e. the result of a consensus among industrial partners.

Summary

Our review of the emerging industry standards for WoT, in particular IPSO, oneIoTa and HAIM, shows that they are built on heterogeneous modeling frameworks with no common ground, although a cross-standard effort exists for their mutual integration. On the other hand, we see the emergence of a consistent cloud of IoT ontologies maintained by the W3C and ETSI and aligned with each other: SSN, SAREF, OM and SEAS. In the following, we will refer to this set of ontologies and their alignment as the *WoT cloud*. We also include the W3C TD model in the WoT cloud, as it is meant to be a bridge between industrial systems and the Semantic Web.

The idea we develop in this paper is to use the WoT cloud as a reference integration model for the semantic integration of IPSO, oneIoTa and HAIM. We introduce next our semantic data integration method.

SEMANTIC DATA INTEGRATION

Semantic data integration, in our case, consists in performing syntactic transformation on data from different source models to a single reference model, such that the semantics of the data is preserved. As highlighted in Sec. 2.2, Semantic Web technologies are a good candidate for this task. Here, we show how arbitrary JSON data described in a schema language (e.g. JSON schema) can be turned into RDF and aligned with the WoT cloud, used as a reference integration model.

As shown on Table 1, all the standards we reviewed are specified in terms of data schemas and meta-models. Roughly, we exploit the former to turn JSON data into RDF and the latter to generate transformation rules expressing alignments between arbitrary data and concepts from WoT cloud.

From JSON to RDF

Our transformation from JSON to RDF relies on the mechanisms specificed by the recent W3C standard called JSON for Linked Data (JSON-LD) [36]. JSON-LD aims at defining a transformation by a *context*, which is a JSON object providing mappings from arbitrary JSON terms to RDF IRIs. In our case, if the JSON terms for a specific standard are known in advance, one can generate a JSON-LD context to map them to minted IRIs and thus obtain an RDF representation of the original JSON document.

More formally, there exists a JSON-LD transformation for any object schema language that can be embedded in first-order logic (FOL). It is trivial to define a mapping function τ to transform any JSON value (i.e. a null, boolean, string, integer, number, array or object value) to a FOL formula. JSON-LD transformation applies to any schema definition S for which a FOL formula Φ exists, such that a JSON value J validates against S if and only if $\tau(J) \models \Phi$. It suffices for this to define a JSON-LD context that maps all atoms in Φ (all fixed terms) to an RDF IRI.

For instance, assuming τ uses the predicates `field`, `key` and `value` to represent JSON objects, the oneM2M example of

Sec. 1 would entail the following FOL formula:

$$\exists x \exists y \exists z(\texttt{field}(x,y) \land \texttt{key}(y,\texttt{containerDefinition}) \land \texttt{value}(y,\texttt{o.o.h.m.binarySwitch}) \land \texttt{field}(x,z) \land \texttt{key}(z,\texttt{powerState}))$$

The corresponding JSON-LD context would be:

$$[\texttt{containerDefinition} \rightarrow \texttt{rdf:type}, \texttt{o.o.h.m.binarySwitch} \rightarrow \texttt{haim:BinarySwitch}, \texttt{powerState} \rightarrow \texttt{haim:powerState}]$$

Given this JSON-LD context, the original oneM2M JSON object would have the following RDF representation after transformation:

```
[] rdf:type haim:BinarySwitch ;
   haim:powerState "false" .
```

Similarly, the OCF object would correspond to the following RDF triples:

```
[] rdf:type oneiota:Binary%20Switch ;
   oneiota:value "false" .
```

This JSON to RDF transformation works for JSON schema, without references (JSON pointers) and regular expressions. To capture the latter two features, monadic second-order logic (MSO) is required [31]. It is likely that one could generalize the transformation formalism to MSO but, since our work was mostly practical and focused on three standards, we leave out this aspect in this paper. All schemas from IPSO, OCF and oneM2M are embeddable in FOL.

Alignment with the WoT Cloud

Given a JSON-LD transformation, semantic integration can be reduced to the problem of semantically aligning RDF terms, like `haim:BinarySwitch` or `oneiota:Binary%20Switch`, with the WoT cloud. The research problem of *ontology alignment* (or *ontology matching*) has been formalized and extensively studied in the past decades [13, 14]. There now exists mature implementations to solve the specific problem of *element-level* ontology alignment. According to the latest benchmarks, some systems consistently perform good on various tasks, with precision results above 75% [9].

However, element-level alignment, which consists in finding semantic relations between individual concepts (e.g. subclass or subproperty relations), fails at capturing more elaborate equivalences involving combinations of concepts. The study of what is often referred to as *complex* ontology alignment is still an emerging field and no approach of the state-of-the-art has proven both efficient and generic yet [14]. We therefore took a parallel path by combining element-level ontology alignment with so-called SPARQL rules to express alignments between WoT standards.

SPARQL rules rely on the SPARQL query language to map a set of RDF statements with variables (`WHERE` clause) to another set of RDF statements (`CONSTRUCT` template), respectively corresponding to the body and the head of a logic rule. A formal equivalence between SPARQL rules and a subset of FOL has been defined [32] and, in parallel, a more practical definition was introduced as part of the SPARQL Inference Notation (SPIN)[11]. The latter is now part of the Shape Constraint Language (SHACL), an official W3C specification [25].

We generate SPARQL rules in a semi-manual fashion in two steps. First, exploiting meta-model information the various source models provide, we manually define rule *templates*, for each meta-model, of the form $\forall\vec{x}\forall\vec{y}\Phi(\vec{x},\vec{y}) \rightarrow \exists\vec{x}'\varphi(\vec{x},\vec{x}')$, where $\vec{x},\vec{y},\vec{z}$ denote the set of variables contained in the FOL formulas Φ,φ. In a second step, the variables of $\vec{y}$ are substituted by concepts from the WoT cloud, on the basis of element-level alignments between atoms contained in Φ and the WoT cloud.

One might note that our rule templates also contain the set $\vec{x}'$ of variables present in the head but not in the body of the rule. Although this is generally regarded as unsafe since inference may never terminate, the algorithm SHACL defines always will: rules can only be applied once. This is not a strict inferencing mechanism, in the sense that it does not define a sound and complete algorithm, but it perfectly suits our needs: since our primary concern is ontology alignment, the rules we generate can never be chained.

As an example, we have the following alignments available for oneM2M:

$$[\texttt{haim:powerState} \cong \texttt{saref:OnOffState}, \texttt{haim:BinarySwitch} \cong \texttt{saref:Switch}]$$

From these alignments, we can construct the following rule for the oneM2M binary switch module:

```
[] a sh:NodeShape ;
sh:targetNode haim:BinarySwitch ;
sh:rule [
  a sh:SPARQLRule ;
  sh:construct
  """
  CONSTRUCT {
    [] a sosa:Observation ;
       sosa:observedProperty [
         a saref:OnOffState
       ] ;
       sosa:hasResult [
         om:numerical_value ?val
       ] ;
       sosa:madeBySensor [
         a haim:BinarySwitch,
           saref:Switch
       ] .
  } WHERE {
```

[11] http://spinrdf.org/

```
    ?container a haim:BinarySwitch ;
               haim:powerState ?val .
  }
  """
]
```

and the following one for an OCF switch, with similar alignments:

```
[] a sh:NodeShape ;
sh:targetNode oneiota:Binary%20Switch ;
sh:rule [
  a sh:SPARQLRule ;
  sh:construct
  """
  CONSTRUCT {
    [] a sosa:Observation ;
       sosa:observedProperty [
         a sosa:ObservableProperty
       ] ;
       sosa:hasResult [
         om:numerical_value ?val
       ] ;
       sosa:madeBySensor [
         a oneiota:Binary%20Switch,
           saref:Switch
       ] .
  } WHERE {
    ?request a oneiota:Binary%20Switch ;
             sw:responses ?data .
    ?data oneiota:value ?val .
  }
  """
] .
```

IMPLEMENTATION & EVALUATION

The two main implementation tasks in this work were the extraction of a JSON-LD context from data schemas (Sec. 3.1) and the computation of element-level alignments (Sec. 3.2). Given a lack of common ground across standards in the data format they use, we implemented context extraction by JavaScript scripting on the JSON and XML files the different standardization bodies provide. For the computation of alignments, we used the AgreementMakerLight (AML) system [15], which reports best performances in the Semantic Web community [9]. We generated SPARQL rules for all model elements provided by the different standards; if no alignment was found by AML, we used generic concepts instead, like `sosa:ObservableProperty` and `saref:Sensor`.

We evaluated our approach on two aspects. First, we looked at the quality of ontology alignment for each SPARQL rule, that is, the level of specificity of the concepts found by AML. We call this aspect *vertical* evaluation (Sec. 4.1). Then, we made a *horizontal* evaluation, by comparing the overlap that exists between standards after data integration (Sec. 4.2). All implementation details (code, data) as well as the results introduced in this paper are also available on Github[12].

[12] https://github.com/vcharpenay/wot-cloud

Table 2. Precision & recall for AML on element-level alignments.

	Precision	Recall
IPSO	26/28 (93%)	26/38 (68%)
oneIoTa	15/17 (88%)	15/31 (48%)
HAIM	23/26 (88%)	23/33 (70%)

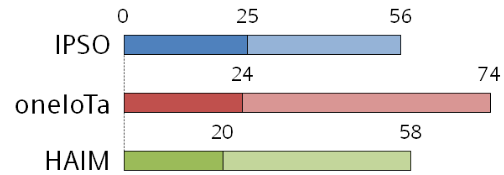

Figure 1. Number of generated rules by standard (darker areas indicate rules include concepts from element-level alignments).

Vertical Evaluation

The correctness of our context extraction scripts can be easily tested by generating random data that validates against some schema and ensure that the generated SPARQL rules for that schema get triggered accordingly. In the following, we focus on the performances of AML.

To be able to evaluate its performances, we manually reviewed the possible alignments with the WoT cloud for all model elements from IPSO (53 LWM2M object definitions, 97 resource definitions), oneIoTa (86 Open API specifications) and HAIM (13 device templates and 41 module classes). For each model element, we provided zero or one alignment. From this ground truth, we computed the precision and recall of AML (Table 2).

In all cases, AML has a very high precision. The rather low recall for oneIoTa is more a sign of implementation peculiarities than theoretical limitations. This case put aside, AML results indicate that our approach can scale up, if soundness is favored over completeness. This is of importance, since the size of the standards models available for WoT is expected to grow. OneM2M, for instance, has planned several iterations during which new module classes will be developed.

In total, we could generate 188 SPARQL rules, as shown on Fig. 1 (our ground truth alignments were also included in the generation process). We defined one rule template for each standard, targeting sensors only. In a large-scale setup, actuator definitions may also be included, using the same appraoch.

Figure 1 also shows that only few rules include concepts resulting from element-level alignments (43% at most, for IPSO). This means that most rules would only produce generic statements, essentially that ‘a sensor made some measurement with a certain numeric value’. This lack of details in the semantics of the measurement indicates a lack of exhaustivity in the WoT cloud: some concepts are missing. There is, for instance, no concept for ‘oximeter’ or ‘audio device’ in SAREF or OM. However, it appears that most concepts missing in the WoT cloud are not shared across standards, so that data integration is of little value for these concepts. Our next evaluation shows, indeed, that IPSO, oneIoTa and HAIM share a small set of concepts only.

Horizontal Evaluation

Evaluating the correctness of the rules we generated requires a reference to compare to. One specificity of the standards we consider in our study is their high terminological homogeneity. There are indeed cross-standard efforts to use the same terms to designate IoT-specific concepts. As we already mentioned, events like WISHI contribute to it. As a consequence, it is reasonable to assume that the lexica used in the specification documents of IPSO, oneIoTa and HAIM (i.e. the set of words they contain) is a good approximate for their semantic content.

On this assumption, we extracted two different lexica for each standard: one including only the labels of the model elements they define and one with the labels and textual descriptions of these model elements. We refer to the second lexicon as the *extended* lexicon.

For our extraction, we used well-known text processing techniques also used in ontology alignment [14]: case normalisation from camel case, dot notation and low dash notation; stop word filtering from a list of the 319 most common English words; synonym expansion using WordNet [29]; word stemming using Porter's algorithm [33]. For instance, from oneIoTa's `AtmosphericPressureSensor`, one would obtain the three words `atmospher`, `pressur` and `sensor`.

On Fig. 2, we show a comparison of the overlap between standards for three different sets: lexicon, extended lexicon and set of rules. For the latter, we consider that rules "overlap" if their heads are identical (`CONSTRUCT` pattern). We discarded rules for which no alignment is provided, since their heads are vacuously identical. First, one can see that the three standards have different levels of documentation: HAIM has almost no further documentation than labels themselves, while oneIoTa is much richer. However, regardless of the size of the sets we consider, we can observe a pattern: all intersections are small in comparison, especially between IPSO and HAIM.

This suggests that the semantic overlap between these standards is lower than originally expected, which also means that they apply to distinct ontological domains. Full semantic integration would require a consistent conceptualization covering the union of these domains, which is challenging. For instance, oneIoTa is the only standard to include environmental measurement (like CO2 concentration) while only HAIM gives a definition of an oximeter (measuring blood oxygen saturation). To our knowledge, there is no well-maintained ontology covering any of these two cases.

Moreover, one can note that the intersection is higher on rules than on lexica. This suggests that most of the commonalities between standards have been captured by alignments with the WoT cloud. Another way of observing it is to have a closer look at the intersections themselves. As an example, we reported in Table 3 the core set of 28 words present in all lexica (with labels only). As one could expect, this core lexicon includes words like `sensor`, `energy`, `temperature` and `switch`. But it is also interesting to note that all model elements which include these words have an alignment with concepts from the WoT cloud.

direction	duration	audio	value
switch	mode	button	temperature
rate	measure	frequency	unit
power	factor	time	energi(y)
percentage	level	generator	activity
input	start	pressure	sensor
current	state	humidity	colour

Table 3. Core intersection of all lexica (suffixes in gray were removed by stemming).

64% of these concepts are from SAREF (and they represent more than 80% of the SAREF vocabulary itself) and, although they count for 8% of the total, all SOSA concepts are present in the set (12 axioms). SSN and SEAS, since they define high-level concepts, are only half-preserved after filtering and most OM concepts, whose scope goes beyond WoT, are not included.

Summary

Our evaluation shows that (1) the set of ontologies from the WoT cloud must be extended to cover the domains the different standards target (less than half of the concepts they define are present in the WoT cloud); (2) between themselves, the standards do not cover the same domains either; (3) for the concepts they all define (what can be referred to as the WoT *core* lexicon), the WoT cloud is a good match, especially SOSA and SAREF.

CONCLUSION

In this paper, we presented a transformation method that mostly relies on recently standardized Semantic Web technologies (JSON-LD and SPARQL rules). It allows arbitrary JSON data to be integrated in a unified RDF model, where reference ontologies exist. In the case of WoT, these ontologies are SSN, SAREF and SEAS, which we collectively refer to as the WoT cloud. However, our evaluation on IPSO, oneIoTa and HAIM shows that the ontology engineering work around WoT is a work in progress. It must be extended to cover more than 50% of the concepts these WoT standards involve.

In any data integration method, as it is the case with our approach, human input is inevitable. However, we minimize this effort by working only on schema languages (for the extraction of a JSON-LD context) and meta-models (for the definition of rule templates). Our evaluation shows that rules can then be generated with high precision and satisfactory recall. These results appeal for experiments at a larger scale.

Once their data models are integrated, the three standards we considered here appear complementary rather than competing. If they were interchangeable, the lexical overlap between them would be higher. We can therefore conclude that, if standardization is paramount in the development of WoT, it is likely that no golden standard will emerge. Semantic Web technologies have therefore an important role to play in that respect.

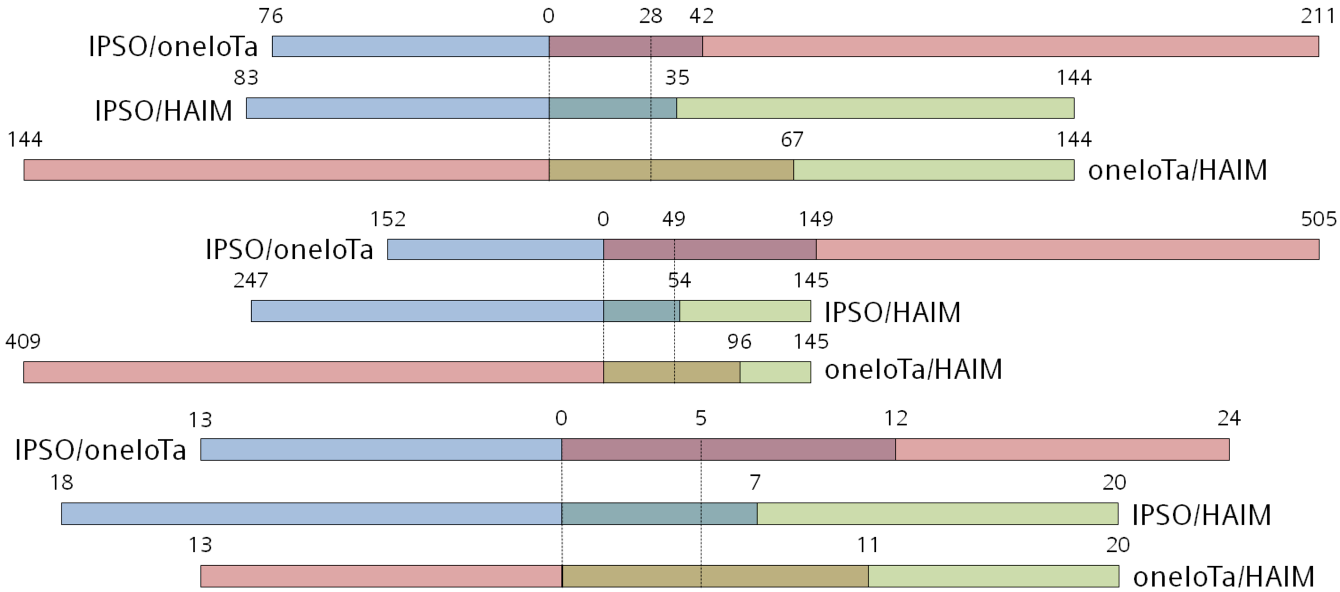

Figure 2. Overlap between standards for different sets: lexicon (top), extended lexicon (middle), set of `CONSTRUCT` templates (bottom).

REFERENCES

1. 1991. *610-1991 - IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. Ieee standard.
2. 2015a. *ITU-T Y.2060 - Overview of the Internet of things*. Itu-t recommendation. International Telecommunication Union.
3. 2015b. *SmartM2M; Smart Appliances Reference Ontology and oneM2M Mapping*. Technical Specification ETSI TS 103 264. European Telecommunications Standards Institute.
4. 2016a. *oneM2M Base Ontology*. oneM2M Technical Specification TS-0012-V-0.7.0. oneM2M.
5. 2016b. *oneM2M Home Appliances Information Model and Mapping*. oneM2M Technical Specification TS-0023-V-2.0.0.
6. 2017. *IPSO Smart Object Guideline*. Technical guideline. IPSO Alliance.
7. 2017a. *Lightweight Machine to Machine Technical Specification*. Oma technical specification. Open Mobile Alliance.
8. 2017b. *OpenAPI Specification Version 3.0.0*. Linux foundation collaborative project. Open API Initiative.
9. Manel Achichi, Michelle Cheatham, Zlatan Dragisic, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Ian Harrow, Valentina Ivanova, and others. 2017. Results of the ontology alignment evaluation initiative 2017. In *OM 2017-12th ISWC workshop on ontology matching* (2017). 61–113.
10. Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The Internet of Things: A survey. 54, 15 (2010), 2787–2805.
11. Victor Charpenay, Sebastian Käbisch, and Harald Kosch. 2016. Introducing Thing Descriptions and Interactions: An Ontology for the Web of Things. In *Joint Proceedings of the 3rd Stream Reasoning (SR 2016) and the 1st Semantic Web Technologies for the Internet of Things (SWIT 2016) workshops* (2016-10-17). CEUR-WS.org.
12. Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. 2012. The SSN ontology of the W3C semantic sensor network incubator group. 17 (2012), 25–32.
13. Jérôme Euzenat. 2004. An API for Ontology Alignment. In *The Semantic Web – ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings*. Springer Berlin Heidelberg, 698–712.
14. Jérôme Euzenat and Pavel Shvaiko. 2007. *Ontology Matching*. Springer Berlin Heidelberg.
15. Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F. Cruz, and Francisco M. Couto. 2013. The AgreementMakerLight Ontology Matching System. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences* (2013). Springer Berlin Heidelberg, 527–541.
16. Carolina Fortuna and Marko Grobelnik. 2013. The Web of Things. ISWC 2013 Tutorials. (2013). `http://iswc2013.semanticweb.org/content/web-things`

17. Dominique Guinard. 2011. A Web of Things Application Architecture – Integrating the Real-World into the Web. (2011).
18. Dominique Guinard and Vlad M Trifa. 2009. Towards the Web of Things: Web Mashups for Embedded Devices. In *Proceedings of WWW (International World Wide Web Conferences)*.
19. D. Guinard, E. Wilde, and V. Trifa. 2010. First IEEE international workshop on the Web of Things WoT 2010: Message from the workshop chairs. In *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (2010-03). 1–2.
20. A. Gyrard, C. Bonnet, K. Boudaoud, and M. Serrano. 2016. LOV4IoT: A Second Life for Ontology-Based Domain Knowledge to Build Semantic Web of Things Applications. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)* (2016-08). 254–261.
21. Armin Haller, Krzysztof Janowicz, Simon Cox, Danh Le Phuoc, Kerry Taylor, and Maxime Lefrançois. 2017. Semantic Sensor Network Ontology. (2017). `https://www.w3.org/TR/vocab-ssn/`
22. Jack Hodges, Kimberly García, and Steven Ray. 2017. Semantic Development and Integration of Standards for Adoption and Interoperability. (2017).
23. C Jennings, Zach Shelby, J Arkko, Ari Keranen, and Caster Bormann. 2017. Media Types for Sensor Measurement Lists (SenML). (2017). `https://tools.ietf.org/html/draft-ietf-core-senml-10`
24. Sebastian Käbisch and Takuki Kamiya. 2017. Web of Things (WoT) Thing Description. (2017). `https://www.w3.org/TR/wot-thing-description/`
25. Holger Knublauch and Dimitris Kontokostas. 2017. Shapes Constraint Language (SHACL). (2017). `https://www.w3.org/TR/shacl/`
26. Matthias Kovatsch, Kajimoto Kazuo, and Michael McCool. 2016. Web of Things Working Group Charter. (2016). `https://www.w3.org/2016/12/wot-wg-2016.html`
27. Maxime Lefrançois. 2017. Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Pattern. CEUR-WS.org.
28. Maxime Lefrançois, Jarmo Kalaoja, Takoua Guariani, and Antoine Zimmermann. 2016. *SEAS Knowledge Model*. Deliverable 2.2. ITEA2 12004 Smart Energy Aware Systems.
29. George Miller. 1998. *WordNet: An electronic lexical database*. MIT press.
30. João Moreira, Laura Daniele, Luis Ferreira Pires, Marten van Sinderen, Katarzyna Wasielewska, Pawel Szmeja, Wiesław Pawłowski, Maria Ganzha, and Marcin Paprzycki. 2017. Towards IoT platforms' integration: Semantic Translations between W3C SSN and ETSI SAREF. CEUR-WS.org.
31. Felipe Pezoa, Juan L. Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. 2016. Foundations of JSON Schema. ACM Press, 263–273.
32. Axel Polleres. 2007. From SPARQL to rules (and back). ACM Press, 787.
33. M. F. Porter. 1980. An algorithm for suffix stripping. 14, 3 (1980), 130–137.
34. Martín Serrano, Payam Barnaghi, Francois Carrez, Philippe Cousin, Ovidiu Vermesan, and Peter Friess. 2015. IoT Semantic Interoperability: Research Challenges, Best Practices, Recommendations and Next Steps. (2015).
35. Zach Shelby, Klaus Hartke, and Caster Bormann. 2014. The Constrained Application Protocol (CoAP). (2014). `https://tools.ietf.org/html/rfc7252`
36. Manu Sporny, Dave Longley, Gregg Kellogg, Markus Lanthaler, and Niklas Lindström. 2014. JSON-LD 1.0 - A JSON-based Serialization for Linked Data. (2014). `http://www.w3.org/TR/json-ld/`
37. Vlad M Trifa. 2011. Building Blocks for a Participatory Web of Things: Devices, Infrastructures, and Programming Frameworks. (2011).