Semantic Model Extensibility in the Context of Interoperable IoT Data Exchange Platforms

Yulia Svetashova
Robert Bosch GmbH, Corporate Research
Renningen, Germany
Email: yulia.svetashova@de.bosch.com

Stefan Schmid
Robert Bosch GmbH, Corporate Research
Renningen, Germany
Email: stefan.schmid@de.bosch.com

***Abstract*—Data exchange platforms and marketplaces are gaining popularity as the "central point of discoverability" [1] for IoT data. They utilize different information models to represent heterogeneous data in a uniform and interoperable manner. Those platforms have the need to dynamically extend and enrich their semantic models in order to accommodate new data offerings. Using the BIG IoT [7] semantic models and the resulting knowledge graph as the basis, we propose a new approach to incorporate user-defined semantic annotations into the model on the fly, which firstly makes them usable before their inclusion into an official release of the model, and secondly minimizes the efforts required from ontology engineers in the model evolution phase. The process of new annotation inclusion is based on the dynamic generation of user interface elements (e.g. web-forms) from annotation patterns stored in the BIG IoT knowledge graph. By filling in user interface forms, a co-operative user creates an unambiguous description of a new concept meaning and connects the concept with related ones, thus, preserving knowledge graph integrity and model consistency.**

I. Introduction

The "development of extensible context models" representing IoT data is one of the key requirements to ensure semantic interoperability in IoT formulated by the European Research Cluster on the Internet of Things [6]. The need to make models *extensible* is a natural concern as new contexts of IoT data usage constantly arise, and new devices, sensor and data types appear. Every data exchange platform starts to operate with an initial information model (schema, ontology[1]) which covers relevant use cases and data samples. In the IoT data exchange contexts, no ontology is complete or has full coverage of all possibly needed semantic concepts, and its life cycle presupposes further enrichment, "timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artefacts" [8] known as ontology evolution process.

Ontology evolution comprises the following sub-tasks: 1) capturing required changes, 2) change representation using a formal language, 3) testing effects of the changes, resolving conflicts and forming a complete change request, 4) change implementation and verification, and finally, 5) change propagation to dependent data and affected applications [2]. The present work builds on the existing ontology evolution methodologies (see the comprehensive surveys on it in [2], [4]). Even though this problem is a traditional topic in the databases, knowledge management and Semantic Web communities, on-the-fly model extension remains rather unexplored.

In the context of IoT data exchange, we define on-the-fly model extension as the ability of a model user, in most cases a data provider, to add new concepts and relations and use newly created model elements to generate semantic descriptions (metadata) for new data offerings. The proposed elements are dynamically added to the model without an ontology engineer intervention, and initially marked by using a special namespace identifier. Later on, an ontology engineer examines the proposed changes and makes one of three decisions: 1) *accept*: add a proposed concept to the model (in the following release the new concept is placed in a corresponding domain model); 2) *replace*: use another concept already defined in the model; 3) *re-model*: introduce a new concept and change the annotation.

Irrespective of the ontology engineer's later decision, the initial proposal operates as the part of the model for a limited period of time, and thus, fulfills the overall goal of making data from new providers machine-readable and usable together. We hypothesize that guiding a user during the process of introducing changes will highly optimize the quality of such open data exchange systems and will ease the subsequent ontology evolution procedure performed by the engineers. This is especially important at the early stages of the operation of an open data exchange system as the demand for model extensibility is high, and the vocabulary used for metadata creation is not familiar to data providers.

The rest of the paper is structured as follows: in Section II, we describe the context of BIG IoT project and its semantic modeling framework. Section III presents the details of the proposed approach to on-the-fly ontology extension. We conclude the discussion and outline the directions of future work in Section IV.

II. Semantic Modeling in the BIG IoT Project

The proposed approach to semantic model extensibility is discussed in the context of BIG IoT project[2] [7] which

[1]From now on, we narrow the space of information models to ontologies, "formalized vocabularies of terms, often covering a specific domain and shared by a community of users. They specify the definitions of terms by describing their relationships with other terms in the ontology" [9].

[2]"Bridging the Interoperability Gap in the Internet of Things", see the project web-site: http://big-iot.eu.

978-1-5090-5873-0/17/$31.00

addresses the interoperability problem creating an IoT data exchange platform – a data marketplace, and developing APIs along with SDKs for data providers and consumers to offer and search for data programmatically. Heterogeneous data being exposed on the marketplace is annotated according to a unified scheme, gaining unambiguous semantics shared by all marketplace participants, and metadata is further stored in a knowledge graph. Uniformly annotated data is a key enabler of cross-platform and cross-domain IoT data integration and collaborative data use, and thus a core of emerging IoT ecosystems.

To set up the context for our approach, we first describe the BIG IoT modeling layer cake which comprises 3 layers: Core, Domain, and Application ontologies. All of them are exposed on the Web as external extensions of the *schema.org* vocabulary. The Core ontology provides concepts that verbalize basic marketplace functionality, its actors (e.g. `core:Provider`[3], `core:Consumer`, `core:Organization`) and their activities (e.g. `core:Offering`, `core:OfferingSubscription`). To describe a data or service offering, a special data structure called *offering description* is derived from the Core model. It is used in the API and in the marketplace web-portal to annotate offerings with the fields, such as `core:name`, `core:price`, `core:license`, `core:endPoint`, `core:category`, `core:spatialCoverage`, `core:hasInput` and `core:hasOutput`.

The last 3 fields of the offering description are the connection points to the Application and Domain models. The Application model contains categories of data offerings sold on the marketplace (e.g. `Mobility`, `Environment`), further subdivided into subcategories (e.g. `Mobility` – `Parking`, `Charging`, `LocationTracking`, and `Environment` – `AirPollution`, `NoisePollution`), as well as links to the expected input/output data types for these categories as defined in the Domain ontologies.

At the current state of the development, the BIG IoT marketplace supports two domain ontologies – Mobility and Environment – contains concepts to characterize the meaning of the exchanged data (see, for instance, the Mobility domain concepts: `mobility:BikeSharingStation`, `mobility:ParkingSite`, `mobility:Accident` and their corresponding input and output data types: `mobility:NumberOfAvailableParkingSpaces`, `mobility:NumberOfAvailableBikes`, `mobility:AccidentType`). Our domain ontologies re-use (when appropriate) concepts of well-known models: *schema.org*, DATEX II, QUDT (Quantities, units, data types ontology), OM (Ontology of measurements), etc. either directly incorporating them by using their URIs or referencing them via the "rdfs:seeAlso" or "dcterms:source" predicates.

To sum up, the BIG IoT semantic models reflect the situation of data exchange: they define an offering's type or categorization, terms of use, as well as the input / output data format and meaning. In order to accommodate new types of data offerings on our marketplace, extensibility is needed especially for the domain modeling layer to describe relevant input and output data.

```
[
  {
    "latitude" : 48.2517561098094,
    "longitude" : 11.637541693635,
    "freeSpots" : 67
  }
]
```

Listing 1. Parking data

We start with the detailed description of how semantic annotation for input and output works using 2 simple data offerings. The first one (see the data sample in the Listing 1) is a service which outputs an array of parking sites with location and availability information. The second service delivers car diagnostics data: fuel consumption and CO_2 emissions (Listing 2).

```
[
  {
    "carId" : "46et9900",
    "consumption" : 12.9686,
    "CO2" : 2.8804
  }
]
```

Listing 2. Car diagnostics data

In order to be able to integrate these data on the fly, a consumer needs to know the context of measurement: which class of real-world objects is observed and characterized by the fields. For the first offering, this will be a parking site, in the second case, a car. The context can be inferred from the category (or subcategory) of the offering which is directly linked to some domain model concept (via the Application model). For instance, the `Parking` category is linked to the concept of `mobility:ParkingSite`; its properties – the characteristics of this real-world object (`mobility:NumberOfAvailableParkingSpaces`, `mobility:ParkingSiteStatus`, etc.), are used to annotate output fields semantically. The `CarDiagnostics` category is linked to the concept of `schema:Car`. For each field, a return value type and a unit of measurement can optionally be specified.

```
Output data
  "fieldName" : semantics
                data type*
                unit of measurement*
```

Listing 3. Field – Annotation graph pair

In the knowledge graph, the default value types and units of measurement (if applicable) can be stored for each property annotation to be used if a data provider didn't specify them

[3] The following namespace prefixes are used in this paper: `core:` – http://schema.big-iot.org/core/; `mobility:` – http://schema.big-iot.org/mobility/; `schema:` – http://schema.org/; `rdf:` – https://www.w3.org/TR/rdf-schema/; `rdfs:` – https://www.w3.org/2000/01/rdf-schema#; `om:` – http://www.wurvoc.org/vocabularies/om-1.8/; `dcterms:` – http://dublincore.org/documents/dcmi-terms/; `sosa:` – https://www.w3.org/ns/sosa/; `qudt:` – http://qudt.org/1.1/schema/qudt; `xsd:` – https://www.w3.org/2001/ XMLSchema; `skos:` – https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html.

explicitly. In the core IoT data use cases, the input / output data sections of the offering description constitute sets of *field – annotation graph* pairs, each of which can be either 1:1 (“field name" – semantic annotation), 1:2 (“field name” – semantic and value type annotations), or 1:3 (“field name" – semantic, value type and measurement unit annotations) relations[4].

```
Output data
  "freeSpots" :  mobility:NumberOfAvailableParkSpaces
                 xsd:int
                 ---

  "consumption" : mobility:FuelConsumption
                 xsd:float
                 om:liter_per_hour
```

Listing 4. Example annotations

To assure consistency of the marketplace knowledge graph, the BIG IoT domain models are aligned with the SOSA (Sensor, Observation, Sample, and Actuator) ontology [3]. From the example outputs above, it is apparent that a substantial part of the output data captures the results of `sosa:Observation`. We re-use the classes `sosa:FeatureOfInterest` (“The thing whose property is being estimated or calculated in the course of an Observation to arrive at a Result” [3]), `sosa:ObservableProperty` (“An observable quality (property, characteristic) of a FeatureOfInterest” [3]), and `sosa:Result` (either a complex object comprising a value and a measurement unit annotations or a simple value, rdfs:Literal) and some of the related SOSA properties to build field annotation graphs (their basic structure is sketched in the Listing 5).

```
sosa:Observation
  [hasFeatureOfInterest ] sosa:FeatureOfInterest
  [observedProperty ]     sosa:ObservableProperty
  [hasSimpleResult ]      rdfs:Literal
  ==========================================
  [hasResult ]       sosa:Result
                          rdf:type qudt:QuantityValue
                          qudt:numericValue
                          qudt:unit
```

Listing 5. Basic annotation graph structure for a input/output field of an offering

In our first example offering (Listing 1), the `mobility:ParkingSite` class is modeled as a sub-class of `sosa:FeatureOfInterest`, its characteristic `mobility:NumberOfAvailableParkingSpaces` – as a sub-class of `sosa:ObservableProperty`, and the result of observing this property as an `rdfs:Literal` of `xsd:integer` data type.

To create annotations for the second offering (Listing 2), the re-used concept `schema:Car` is subsumed by `sosa:FeatureOfInterest`, and its properties `mobility:FuelConsumption`, `mobility:EmissionsCO2` are the subclasses of `sosa:ObservableProperty`, etc. In the next section, we show how the described alignment mechanism can be used to optimize the inclusion of the new concepts proposed by a user into domain ontologies.

III. Semantic Model Enrichment Approach

We explain our on-the-fly model extension approach based on a concrete example. Let us assume that a user provides a description of his new offering via the marketplace web-portal and starts with the selection of the offering category: `Parking` and subcategory: `ParkingSite`. The user then defines the semantic annotations for the input and output data. However, the drop-down menu of the expected output annotations does not yet offer any adequate semantic term for the data property: “totalCapacity". To overcome this limitation, the user can propose a new semantic term to annotate the property. In response to a click on the “Propose your own" choice, the web UI is dynamically extended with a set of forms: (radio) buttons and editable text input fields with labels generated from the basic annotation graph explained above. The dynamically extended web form involve the user to actively contribute to the semantic model extension in a user friendly manner. I.e. the system takes into account previously made user choices to limit the user involvement to the necessary inputs.

In Table 1, we show how the web-portal automatically generates forms for user input based on the basic annotation graph triples[5] from Listing 5. The user-provided inputs are subsequently transformed into a set of triples which both annotate the field named “totalCapacity" in the offering, and create a new subclass of `sosa:ObservedProperty` `proposed:TotalCapacity` which is related to `mobility:ParkingSite`, a subclass of `sosa:FeatureOfInterest`. The return value type characterization is performed in an analogous manner. I.e. if a unit of measurement is relevant for a `sosa:Result` of a proposed `sosa:ObservedProperty`, a drop-down list with concrete units from the ontology of units of measure (OM) is generated in the web-form and the user is asked to select the matching unit.

With the help of a basic annotation graph, we can involve the user in the process of semantic model extension and automatically generate targeted web-forms, allowing users with minimal extra effort to contribute to the knowledge creation. The most notable benefit of this approach is that a newly proposed semantic annotation is semantically enriched through established relations to the existing concepts.

Swiss linguist and semiotician Ferdinand de Saussure stressed that signs are always parts of complex systems, and their meaning is defined by the relations between signs in such

[4]Exposing other types of data (statistical or aggregated data, for instance) may require more complex modeling. See, as an example, the LBASense Analytic API (http://www.dfrc.ch/wp-content/uploads/2016/05/LBASense-Analytic-API-V3.0.pdf) where the visit duration data showing the number of visitors broken down into fixed length interval values. It contains fields like: “from1To5Minutes", “from5To10Minutes", “from2To3Hours", etc. Here, to annotate data semantically, not only a concept of *number of people* is to be modeled, but a *visit duration interval*, with *placeholders* for values and time units.

[5]Note that questions in natural language forming the UI forms’ labels transform the cited above definitions of corresponding SOSA concepts stored as the content of the `skos:definition` property (see: https://www.w3.org/ns/sosa/).

Step	Known Triple(s)	Generated Label and UI Element	User Choice and Derived Triple(s)
(1)	[previous choice:] Observation/id1 __sosa:hasFeatureOfInterest ____mobility:ParkingSite . [question to solve:] Observation/id2 __sosa:hasFeatureOfInterest ____?Feature .	Does the data output relate to mobility:ParkingSite? Yes No [radio buttons]	[user selects "Yes":] Observation/id2 __sosa:hasFeatureOfInterest ____mobility:ParkingSite . [proceed to (3)] [user selects "No":] [proceed to (2) to generate new label and UI element]
(2)	[question to solve:] Observation/id2 __sosa:hasFeatureOfInterest ____?Feature .	Please select the thing which property is being measured. [show drop-down list of subclasses of sosa:FeatureOfInterest within the Mobility domain]	[user selects "mobility:ParkingSpace":] Observation/id2 __sosa:hasFeatureOfInterest ____mobility:ParkingSite .
(3)	[previous choice:] Observation/id2 __sosa:hasFeatureOfInterest ____mobility:ParkingSite . [question to solve:] Observation/id2 __sosa:ObservedProperty ____?Property .	Which quality (property, characteristic) of mobility:ParkingSite is being measured? [show text field]	[user enters "TotalCapacity":] proposed:TotalCapacity __rdf:typeof ____ssn:Property, ____sosa:ObservableProperty; __ssn:isPropertyOf ____mobility:ParkingSite . Observation/id2 __sosa:ObservedProperty ____proposed:TotalCapacity .

Table I
Dynamic web-forms generation and processing

systems [5]. The `sosa:Observation`-centric relations we have used are an example of *syntagmatic relations* in Saussure's terminology. Syntagmatics is based on co-occurrence of signs and helps humans to detect properties which operate together to create meaning. Growing the knowledge graph, we are able to capture that `sosa:ObservedProperty` Temperature is measured in `qudt:unit` Celsius, Fahrenheit or Calvin; that `sosa:ObservedProperty` Speed refers to a `sosa:FeatureOfInterest` Vehicle and its `qudt:unit` is kilometer per hour, etc. These regularities can further be incorporated (through inference) into the new semantic model extension process. For instance, provided that the `sosa:Result` unit for the proposed semantic annotation is specified as Revolution per minute, a dynamically generated web-form can automatically suggest `sosa:ObservedProperties` and corresponding `sosa:FeaturesOfInterest` (e.g. car or washing machine), which have this unit as part of their basic annotation graph. Syntagmatic relations put similarly structured `sosa:Observation` situations closer to each other, and thus can help users to correct semantic annotations via the web-portal as the model grows. E.g. a user who introduced a new semantic annotation in the initial stage due to the lack of adequate terms may subsequently correct the annotation, as the web-portal displays closely related concepts next to the term initially selected, and thus making the change really easy.

Another type of relations between signs defined by F. de Saussure is called *paradigmatic*. It groups signs that can substitute each other, so they define semantically similar things. A very straightforward application of this idea might be, for instance, exposing a user to all `sosa:ObservedProperties` of a certain `sosa:FeatureOfInterest` after it is selected. A user might also be asked which of the properties is the most similar semantically to a proposed one, and based on the answer stored as triple, the new annotation gains a more precise characterization. Going back to our example in Table 1, instead of the free-text form and label shown in Step (3), the following selection list could be first generated: "ParkingSitePriceCategory", "ParkingSiteAvailabilityStatus", "ParkingSiteOpeningStatus", "NumberOfAvailableParkingSpaces". It is very likely that a co-operative user would select "NumberOfAvailableParkingSpaces", and as a result might also consider the naming conventions of the current annotations and in turn propose a new annotation like "TotalNumberParkingSpaces".

The proposed approach to guide a user through the new concept inclusion process relies on syntagmatic and paradigmatic relations of concepts and uses basic annotation graphs to unify the description. The approach brings the following benefits: first, it allows to contextualize the user-proposed annotations, to place it (assuming a co-operative user) closely to the semantically related concepts, and it assures more coherent and less ambiguous description of it. Second, it reduces the risk of concept duplication and unconventional naming. Third, answering questions guided by the basic annotation graph creates a ready-to-use description which can be used in UIs (web-forms) and shared with others as linked data. Another remarkable benefit is that newly proposed concepts can simultaneously be used for offering metadata generation. Due to the links to the related structures, the semantic annotations of the input / output data is partially known and machine-processable even before an ontology engineer approves it.

The proposed approach is applicable and beneficial in the context of data exchange platforms in general: it means more clarity and less efforts for data-providers when creating offering descriptions, and for ontology engineers during model evolution and maintenance. For data consumers the proposed approach is also favorable, as more meaningful offering descriptions are provided and consumers can leverage the extended models also for defining their offering queries.

IV. Conclusions and Future Work

We have introduced an approach to include new user-defined semantic annotations for data offering input/output fields, and thus, allow on-the-fly extension of the semantic model supported by a data exchange platform. The proposed approach to semantic model extensibility explicitly imitates substantial aspects of the ontology engineer decision making process by following uniform consistent modeling principles and leveraging concepts' relations. Though filling-in dynamically generated UI elements is still a manual process from the user side, the underlying structure captured by basic annotation graph is a step towards partial concept inclusion automation which is the primary direction for our future work.

We have described the approach referring for simplicity to only one data pattern related to the sensor measurements as it is the core use case for most IoT related data exchange platforms. With this, most of the IoT streaming data descriptions can be fully aligned with the SOSA ontology framework. Still, for some of the spatial and temporal aspects of data offerings different semantic annotation patterns are appropriate. Moreover, there exist drastically dissimilar patterns of data representation (statistical data, historic datasets) and types of services (forecasting, reservation and payment). Also, the typology of data offerings is still to be studied and corresponding basic annotation graphs are to be developed.

Finally, we plan to investigate new types of concepts' relations which have the potential to ease new concept inclusion – both user-mediated and automated – and further explore the flexibility and dynamic interoperability of graph-based models in the context of IoT data exchanges.

Acknowledgment

This work has been developed in the project "Bridging the Interoperability Gap" (BIG IoT) funded by the European Commission's Horizon 2020 research and innovation program under grant agreement No 688038.

References

[1] J. Deichmann et al., *Creating a Successful Internet of Things Data Marketplace*. McKinsey Quarterly 10/2016.

[2] G. Flouris et al. *Ontology Change: Classification and Survey*. Knowledge Engineering Review, Vol. 23. (2008): 117–152.

[3] A. Haller at al., *Semantic Sensor Network Ontology*. W3C Recommendation. W3C. https://www.w3.org/TR/vocab-ssn/.

[4] A. M. Khattak et al., *Ontology Evolution and Challenges*. Journal of Information Science and Engineering. 29.5 (2013): 851–871.

[5] F. de Saussure, *Cours de linguistique generale*, published by C. Bally and A. Sechehaye. Paris: Payot, 1995 (1916).

[6] M. Serrano et al., *IoT Semantic Interoperability: Research Challenges, Best Practices, Recommendations and Next Steps*. IERC, 2015.

[7] S. Schmid et al., *An Architecture for Interoperable IoT Ecosystems*. International Workshop on Interoperability and Open-Source Solutions. Springer, 2016.

[8] L. Stojanovic, *Methods and tools for ontology evolution*. PhD Thesis, University of Karlsruhe, Germany, 2004.

[9] W3C OWL Working Group, *OWL 2 Web Ontology Language Document Overview*. 2009.