Semantic Interoperability and Interrater Agreement in Annotation of IoT Data

Yulia Svetashova[1,2][0000-0003-1807-107X], Stefan Schmid[1], and York Sure-Vetter[2]

[1] Robert Bosch GmbH, Corporate Research,
Robert-Bosch-Campus 1, 71272 Renningen, Germany
{yulia.svetashova,stefan.schmid}@de.bosch.com
[2] Karlsruhe Institute of Technology, AIFB,
Kaiserstr. 89, 76133 Karlsruhe, Germany
york.sure-vetter@kit.edu

Abstract. Data management has become a critical ability in today's data-driven businesses. In the Internet of Things (IoT) domain, sensors, devices and applications generate huge amounts of data. To take advantage of this data, new storage and exchange solutions, e.g., IoT data repositories, enterprise data lakes and IoT data marketplaces, are emerging. These emerging data storage and exchange solutions allow authorized users to discover and access heterogeneous data streams and integrate them across stakeholder boundaries. These new requirements challenge the typical function of metadata, namely, to ease data discovery and identification; its usage as a mediator for data integration comes to the fore.

We present an approach to annotate IoT datasets by mapping their schemata to corresponding ontology terms, paired with an on-the-fly ontology extension mechanism based on templates. We further introduce a framework to evaluate semantic interoperability in this complex setting via the agreement between domain experts on a set of annotation-extension tasks. We finally outline where this evaluation framework can be used to improve the system iteratively and to avoid potential semantic interoperability conflicts.

Keywords: Semantic interoperability · Semantic annotation · Ontology extension · Interrater agreement · Internet of Things.

1 Introduction

Data has become a strategic asset nowadays. As has been acknowledged by M. Chui et al. [7], one of the main bottlenecks in making it accessible and usable is the lack of high-quality data labeling support. Modern data storage and exchange solutions (e.g., data catalogues, data marketplaces and enterprise data lakes) allow easy access to various types of data, as well as integration of data into analytic workflows, and collaboration of data scientists and business users over data. However, to enable these intelligent functionalities, the submitted datasets should be semantically interoperable.

Semantic interoperability is defined as the ability of two or more systems to interpret the content and meaning of the exchanged information [17]. In the context of the Internet of Things (IoT), it is typically achieved by adding semantic annotations to the raw data or by *mapping* the diverse schemata of each data source *to a unified representation*: an ontology, a taxonomy, or a controlled vocabulary (see [2]).

The usage of a shared ontology or a set of ontologies as “the best pathway to achieve semantic interoperability” [16] was tested in several European research projects: OpenIoT [28], IoT Lite [4], FIESTA-IoT [3], BIG IoT [6] and other. Among the challenges and requirements, related to interoperability, these projects report that a shared ontology must be evolvable over time as new data sources, sensors or IoT devices appear.

Traditionally, ontology is extended in a centralized manner: data providers or device owners first submit requests for extension and then use the updated ontology for annotation. In practice, it means long waiting times before the ontology term is available to perform a task in question. Multiple rounds of communication between data providers and ontology engineers take place to clarify the meaning of the requested ontology terms[3]. To address this limitation, several approaches to involve data providers in the ontology extension process were suggested (see, e.g., [20], [18]).

The core characteristic of these approaches is the *coupling of the annotation process with the on-the-fly ontology extension*. When a data provider lacks a term to perform an annotation task, s/he can use a simple procedure to add it via the graphical user interface (GUI). The Schema Editor of OpenIoT middleware infrastructure [20] is an example of an implemented system. It is a Web-based application, which targets non-ontology expert users. It allows the addition of new sensor types to the SSN-based OpenIoT ontology and their later use in defining descriptions of sensor instances. In both cases, the users provide values via the Web forms. These forms serve as *templates* that guide the user, requiring them to supply information and linking it to the corresponding superclasses. The tool, thus, preserves the ontological foundations of the underlying model.

The decentralization of the ontology extension process might become truly challenging for the external applications that consume newly introduced terms or data which has been annotated with them. If a term description is not accurate or not complete, it can cause incompatibilities in data semantics and structures between the systems, phenomena termed *semantic interoperability conflicts* by J. Park [27]. As an example, imagine that a navigation application expects (in accordance with the specification/description) an input in newton-seconds, while the other system delivers it (contrary to the description) in foot-pound-seconds. In 1999, such a conflict (*data-unit* conflict in Park’s classification) led to the wrong trajectory computation and sent the NASA’s Mars Climate Orbiter spacecraft fatally close to the surface of the planet Mars[4], where it simply burned up.

In the complex setting of annotation with the option to dynamically extend the shared ontology, semantic interoperability conflicts can result either from 1) incorrect mappings of elements of annotated data sources to the ontology terms, or 2) incorrect/incomplete descriptions of newly introduced terms. In both cases, “the process of converging to a uniform semantics can be influenced but not controlled” [30].

In order to build decentralized systems with the dynamic ontology extension component, it is crucial to understand potential causes of semantic interoperability conflicts. When they are known, one can *influence* the extension process. Numerous studies in the bio-medical domain (see, e.g., [22] and related work there) showed how the task of an-

[3] Here we refer to the experience of the team responsible for the maintenance of the ontology in the BIG IoT project. This challenge is also valid in the context of describing datasets in the scientific communities (see [18]).

[4] https://www.jpl.nasa.gov/missions/mars-climate-orbiter/.

notation with the existing terms can be explored via the interrater agreement metrics on a set of tasks addressing potential semantic interoperability conflicts. Our focus stays, however, on the extension component and its impact on the overall agreement. To assess this, we conducted an experiment where 23 non-ontology experts annotated 7 samples of IoT data (56 data points in total) with the terms of a shared ontology and simultaneously extended it if the needed terms were missing.

This paper makes the following contributions:

1) it suggests an experimental setup to explore the complex annotation-extension setting, which consists of the application to annotate IoT data with an ontology extension module, an ontology for IoT data and a set of SOSA[19]-inspired templates, which cover the main types of extensions;

2) it introduces the modeling of the proposed extensions which is used along with the selected existing ontology terms to measure interrater agreement in the annotation-extension task;

3) it presents the results of a user experiment and proposes the interpretation of agreement scores as signals of the potential semantic interoperability conflicts;

4) it outlines the measures to iteratively improve the system based on the findings.

The rest of this paper is organized as follows. Section 2 discusses the relevant research. Section 3 presents our approach to ontology extension implemented as a part of IoT data annotation environment. Sections 4 and 5 provide the details of the experiment with domain experts and discuss the results. Section 6 concludes this article with a summary of accomplished and future research directions.

2 Related work

Previous research relevant for our work can be categorized into three broad areas. Firstly, our work goes in the direction of **collaborative frameworks and tools** used for knowledge engineering and ontology development. Our target users have little (if any) exposure to ontologies, so it is desirable to hide the complexity of the latter. Thus, the most closely related implemented systems are the the Linked Earth Framework [18] and the Schema Editor of OpenIoT [20].

The Linked Earth Framework supports the paleoclimate community to describe their datasets and enable users to add new metadata properties. It employs an initial core ontology and a set of extensions (called proxies), which serve as the basis of the crowd vocabulary. Proposed new metadata properties are available to the community at the time of creation, and they can later decide to include them into the core ontology. In contrast, our tool does not involve community discussions and voting procedures. At some point, ontology engineers approve proposed extensions. Before that the terms are available to the users, but have a special namespace.

The Schema Editor also combines the extension of ontologies (the Sensor Type Editor) with using them to annotate instance data (the Sensor Instance Editor). This is one of the earlier tools for the annotation of the IoT data, focused on describing sensor types by providing their names and observed properties with accuracy and frequency values. As will be shown in Section 3, in our tool we use a different, but also SSN/SOSA-inspired [19] set of templates and a more elaborate characterization of the new ontology terms.

Secondly, we built our system on the ideas coming from the **template-based ontology extension tools**: TermGenie [9], Webulous [21], OTTR [13]; Protégé plugins supporting OPPL [11] and MappingMaster [26]. In those tools, ontology design patterns are used to generate templates. Each template represents a boilerplate for the ontology entity of some type. Domain experts and data curators populate these templates by filling in GUI forms or spreadsheets. Their input is further transformed into the axioms of the shared ontology.

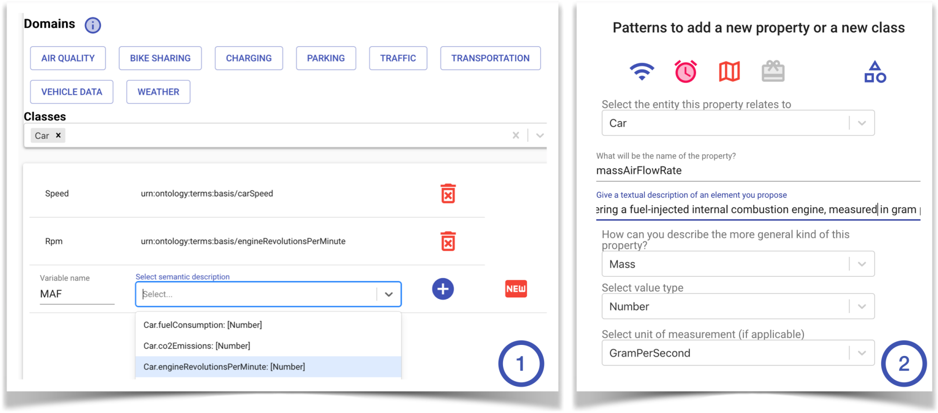
Template-based tools consist of four generic components: 1) a templating mechanism, 2) an input GUI, 3) a template instantiation/expansion processor, and 4) (optionally) an input validation system. We implement these components in our Web-based system prototype. We also add prompts in natural language to the corresponding GUI forms (similar to [23]). Most importantly, we address one research gap of the template-based systems. To the best of our knowledge, there is no systematic work on the usage of templates. Hence, in our experiment, we will try to gain insights on how templates and sets of templates work in practice.

Finally, our experiment design can be compared with the recent study [29], which explores the **agreement of experts** on a task of classifying entities in domain ontologies under upper ontology classes. We take into account the literature on crowdsourced data annotation [10], which models interrater agreement and disagreement, but for these initial experiments we use easily interpretable metrics, such as percent agreement and Fleiss' kappa statistic [12].

In the following we describe our experimental setup and the conducted experiment.

3 IoT Data Annotation Environment

With the objective of exploring semantic interoperability issues in the context of IoT data annotation, we built a system prototype where we extended an interface, developed in the context of the project BIG IoT [6], with a module, which enables experts to add new ontology entities and use them for data annotation (Figure 1).



Classes: Car ×

Speed urn:ontology:terms:basis/carSpeed

Rpm urn:ontology:terms:basis/engineRevolutionsPerMinute

Variable name: MAF

Select semantic description: Select...

Car.fuelConsumption: [Number]

Car.co2Emissions: [Number]

Car.engineRevolutionsPerMinute: [Number]

Select the entity this property relates to: Car

What will be the name of the property? massAirFlowRate

Give a textual description of an element you propose: ering a fuel-injected internal combustion engine, measured in gram

How can you describe the more general kind of this property? Mass

Select value type: Number

Select unit of measurement (if applicable): GramPerSecond

Fig. 1. Annotation environment for IoT data. ① User interface to create mappings. ② User interface to extend ontology by filling in templates.

Suppose we need to annotate the following piece of car diagnostics data {“speed” : 11.75, “Rpm” : 814.12, “MAF” : 7.86, “ts”: “2017-09-19T23:00:00Z”}. The annotation process is as follows: a user first selects a class (e.g., “Car”) and then – its corresponding properties from the dropdown lists (Figure 1.①).

In Figure 2.1, we show how these properties are modeled in the shared ontology. Relation to a class is specified by the property “schema:domainIncludes”. Property “schema:rangeIncludes” characterizes the value type: in this case the float value is mapped to the “schema:Number” data type.

```
basis:carSpeed rdf:type Property;
    rdfs:label "car speed";
    rdfs:comment "The vehicle speed, measured in kilometer per
       hour.";
    schema:domainIncludes basis:Car;
    schema:rangeIncludes schema:Number;
    meta:entityType meta:ObservableProperty;
    meta:propertySematics meta:Speed;
    meta:unit unit:KilometrePerHour .
```

```
TEMP rdf:type rdf:Property;
        rdfs:label _text_;
        rdfs:comment _text_;
        schema:domainIncludes {} ;
        schema:rangeIncludes {} ;
        meta:entityType meta:ObservableProperty;
        meta:propertySematics {} ;
        meta:unit {} .
```

Fig. 2. 1. Property “carSpeed”, Turtle serialization. 2. Template for the *Observable property* pattern, informally introduced.

Additionally, each ontology entity has some characterization in terms of a meta-model. This layer reflects core modeling principles of a shared ontology, which are especially important for extension. For the core use cases we have preferred to use the simple lightweight RDF(S) ontology. Due to having the meta-model, we can always generate a SOSA-compliant representation of an annotated dataset.

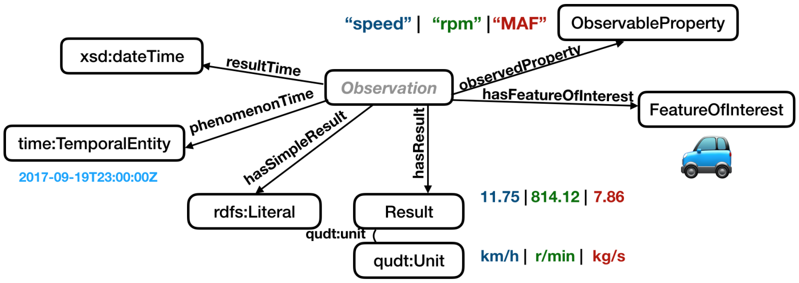


Fig. 3. The sosa:Observation pattern (adapted from [19]) and the car diagnostics data example.

As the basis of our shared ontology we use the *sosa:Observation* pattern [19] , [15]. In our approach, all entities in the ontology have an associated type: “meta:ObservableProperty”, “meta:TemporalProperty”, “meta:SpatialProperty”, or “meta:Class”.

Observable The entity type determines a set of meta-predicates, which specify its semantics. For example, the property “basis:carSpeed” as an instance of “meta:ObservableProperty” has two meta-predicates: “meta:propertySemantics”, related to the higher-level semantics (“meta:Speed”, “meta:Mass”, “meta:Amount”, etc.), and “meta:unit”, specifying units of measurements. Possible objects for those predicates form closed sets.

Temporal Temporal properties have three meta-predicates. The first meta-predicate: “meta:temporalSemantics” models the distinctions from Allen’s [1] seminal work on temporal modeling: time vs. duration, instance vs. interval. The second predicate: “meta:-serviceSemantics” indicates whether the value of the temporal property is measured or predicted. The third meta-predicate: “meta:associatedObservableProperty” is related to the modeling of a “sosa:phenomenonTime”. In the car diagnostic data above, the timestamp value with the key “ts” is related to all observable properties in the dataset. Another option was to relate it to a certain observable property or a subset of properties in a sample.

Spatial Spatial properties with their two meta-predicates conclude the list. Firstly, we can specify the location of an object as a postal address, coordinates alone or a geometry, with many finer distinctions. These options form the possible objects of the predicate “meta:spatialSemantics”. The meta-predicate “meta:structure” handles differences between flat {“lng” : 43.2, “lat” : 9.3} and nested {“location” : {“lng : 43.2223”, “lat” : 9.3 }} structures.

In the following, we explain how the template-based ontology extension works. Each entity type – “meta:ObservableProperty”, “meta:TemporalProperty”, “meta:SpatialProperty” and “meta:Class” – have corresponding extension templates. In Figure 2.2, you can see a skeleton of an observable property. By answering questions in the user-interface forms (Figure 1.②), a user literally fills the gaps in the proposed elements’ description. A user selects options from the dropdown lists, which represent possible objects of the meta-predicates for this property. It results into a structured description of a new element. Most importantly, a new element is linked to the existing ontology concepts, in accordance with the modeling principles of the shared ontology.

Wildcard In addition to the supported entity types, there exists also the “Wildcard” template. This can be used when none of the more expressive templates fits.

4 Towards a Framework to Explore Semantic Interoperability

4.1 Research Questions

Semantic interoperability of the annotated data depends to a large extent on how annotators interpret the underlying shared ontology and a set of templates. Designing this experiment, we wanted to investigate how the option to extend ontologies on the fly would influence the results of the annotation task, how the set of templates worked as a system and what can be said about the quality of a single template based on the interrater agreement. We stated the following research questions:

1 Will the fine-grained modeling of proposed ontology extensions significantly influence the overall agreement among experts in the annotation task?

2 Can we detect and categorize various types of conflicts between the existing ontology entities and the proposed elements via the interrater agreement scores?

3 Can we detect and categorize various types of possible semantic interoperability conflicts by inspecting the interrater agreement scores for the modeling choices made while introducing new elements?

4.2 Experimental Setup

To assess general tendencies and obtain some quantitative estimates for the complex annotation-extension task, we organized an experiment with 23 participants. All of them were computer scientists and engineers from various fields. We have not collected any demographic data and processed the responses as anonymous.

The participants of the study annotated 7 IoT data sources[5] (see Table 1) in JSON format, each of them with a description in natural language, which contained the information required to annotate all the key–value pairs.

Table 1. Table: Datasets used for annotation.

ID	Domain	Number of samples	(intended) Templates to use	Research questions
1	parking	5	–	**1**, **2**
2	street incident	11	–	**1**, **2**
3	air pollution	6	Observable	**1**, **2**, **3**
4	car diagnostics	7	Observable	**1**, **2**, **3**
5	charging	14	Observable, Wildcard, Temporal	**1**, **2**, **3**
6	weather	9	Wildcard, Spatial	**1**, **2**, **3**
7	transportation	4	Temporal, Class	**1**, **2**, **3**

The datasets were presented to the participants in increasing order of complexity. We started with two confidence-building examples, where all needed properties were present in the shared ontology. Then we gradually introduced datasets, which required more complex modeling decisions: nested hierarchies, infrequent data value formats (e.g., duration in the ISO 8601 (“P0DT0H1M12S”) or Unix epoch time), introducing new properties and classes. The last and most complex dataset represented the output of a service (predicted waiting time for a bus), which was very different from the core use case: recorded sensor measurements at a particular timestamp.

Prior to the experiment, all participants watched an introductory video, where we explained the context, showed the annotation environment and annotated one data source, which required adding several terms to the shared ontology.

In the next section, we introduce the relevant interrater agreement metrics and discuss the results of the experiment.

5 Results and Discussion

5.1 Interrater Agreement Measures

Interrater agreement is defined as “the degree to which two or more raters achieve identical results under similar assessment conditions” [24]. It can be calculated at the level of single rated items. For example, the first variable “parkingSpaceId” in Data source 1 was annotated as “schema:identifier” by 20 participants out of 23 (86.95%), the second used element was “basis:operatorIdentifier” selected by 2 participants (8.7%), and the third – “schema:description” used once (4.35%). We will refer to this metric as *percent agreement* on the 1^{st}, 2^{nd}, and 3^{rd}, etc. most frequent choices for the data item.

In addition, we want to obtain a quantitative estimate of experts’ agreement on a set of tasks for the whole experiment. We achieve that by treating shared ontology elements as nominal categories, which are assigned by experts to the rated data items. For

[5] These data sources and their descriptions are available in the following Github repository: https://github.com/YuliaS/PatternBasedExtension.git. In this paper, we discuss mostly the results of introducing new properties.

categorical data, the most common interrater agreement metric is a kappa-type statistic which measures the observed level of agreement between raters for a set of nominal ratings ($\bar{P}$) and corrects for agreement that would be expected by chance ($\bar{P}_e$) [12]:

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{1}$$

J. Cohen [8] was the first to propose kappa for two raters. In our experiments, we use the Fleiss' [12] kappa metric generalized to any constant number of raters.

We base our interpretation of kappa values on the Landis and Koch (1977) [25] benchmark scale. It describes relative strength of agreement associated with kappa ranges, with values from 0.0 to 0.2 indicating *slight* agreement, 0.21 to 0.40 indicating *fair* agreement, 0.41 to 0.60 indicating *moderate* agreement, 0.61 to 0.80 indicating *substantial* agreement, and 0.81 to 1.0 indicating *almost perfect* agreement. Statistics $\kappa < 0.00$ is a *poor* agreement, and $\kappa = 1$ indicates *perfect* agreement.

5.2 Interrater Agreement and Ontology Extension: Tendencies

To compute the Fleiss' kappa statistic, we transform our results' table into a 56×23 matrix in which the columns represent the different raters (n = 23), and the rows represent data items (N = 56), which the raters have characterized either by existing ontology terms or proposed extension elements. We treat both as nominal categories (Table 2). The Fleiss' kappa computed for this matrix was 0.638[6] for the number of categories k = 218, which indicates substantial interrater agreement.

Table 2. Results' table for computing the Fleiss' kappa statistic (a fragment).

user → sample ↓	user01	user02	<...>	user23
1.0	schema:identifier	schema:identifier	...	basis:operatorIdentifier
1.1	basis:parkingSpace-Availability	basis:parkingSpace-Availability	...	proposed:status
...	...	...	...	...
7.3	proposed:timeToArrival	schema:description	...	proposed:busArrivalTime

However, using labels for the proposed elements directly introduced additional 178 categories to the 40 existing ontology elements used for annotation. These labels are not generalizable in the sense that they do not reflect the underlying modeling decisions. Thus, to explore the tendencies in the overall agreement, we need to perform several data transformations.

In Table 3, we show the choices for the element "proposed:status" used to annotate data sample 1.1 by *user23*. These choices come from the closed sets of options displayed in the drop-down lists of the user interface. By combining the options into a sequence, e.g. "Observable_ParkingSite_Text_Quality_None", we obtain a compound name for the proposal, which will be reused for the proposal coined by another user, if the choices were identical. Note that it is possible to represent just the pattern type

[6] Statistics were done using R 3.5.0 (R Core Team, 2018), the *irr* (v0.84.1; Gamer, 2019) and the *reshape2* (v1.4.3; Wickham, 2007) packages.

Table 3. Prompts used as form labels in the *Observable property* template and the user choices for the data sample 1.1 “status” by *user23*.

Select pattern	Select entity this property relates to	Select value type	How can you describe the more general kind of this property?	Select unit of measurement (if applicable)
Observable property	basis: ParkingSite	schema:Text	meta:Quality	meta:None
pattern	*domain*	*range*	*meta1*	*meta2*

(“ObservableProperty”) or pattern and a value type (“ObservableProperty_Text”) and check the expert agreement on a subset of choices. Now, we can substitute the labels of the proposed elements in the results’ matrix (e.g. “proposed:status” → “Observable_ParkingSite_Text_Quality_None”) with the compound proposals’ characteristics and assess the overall agreement.

To answer Research Question **1** – will the fine-grained modeling of proposed elements significantly influence the overall agreement among experts in the annotation task? – we applied the above mentioned substitutions to our initial result matrix. We thus created 12 datasets where the characteristics of the proposed entities were modeled with different levels of granularity.

Table 4. Overall agreement between experts on a set of annotation tasks.

	Granularity: user choices considered	# categories (k)	Kappa
01	none: all proposals labelled “proposed”	41	0.773
02	*pattern*	44	0.718
03	*pattern*, *domain*	82	0.702
04	*pattern*, *range*	61	0.703
05	*pattern*, *domain*, *range*	114	0.691
06	*pattern*, *meta1*, *meta1*	98	0.673
07	*pattern*, *domain*, *meta1*	115	0.673
08	*pattern*, *domain*, *meta2*	112	0.682
09	*pattern*, *range*, *meta1*	89	0.672
10	*pattern*, *range*, *meta2*	84	0.683
11	*pattern*, *range*, *meta1*, *meta2*	125	0.661
11	*pattern*, *domain*, *range*, *meta1*, *meta2*	171	0.652
12	*pattern*, *domain*, *range*, *meta1*, *meta2*, *meta3*	182	0.645

As a baseline, we constructed a dataset where all labels of the proposed elements were replaced by one label “proposed”. This dataset roughly approximates the setting when only existing elements of shared ontology are used to annotate the datasets.

More fine-grained modeling of the user choices introduced additional categories. Their number ranges from 4, when only pattern types (“Observable property”, “Temporal”, “Spatial”, “Wildcard”) were specified, to 142, when all possible combinations of options were considered.

The highest agreement, $\kappa = 0.773$, was obtained for the baseline dataset, the second largest value, $\kappa = 0.718$ – for the dataset, where only pattern types were modeled. The difference between the 1^{st} and the 2^{nd} highest kappa statistics is larger than in any subsequent pairs, where the κ values decrease almost linearly with the increase in the number of categories.

The results in Table 4 show that irrespective of the number of user choices considered, the Fleiss' kappa values remain in the range of substantial agreement scores. Therefore, the overall agreement score for the annotated dataset is influenced by the fine-grained modeling of proposed elements, but not to the extent that it changes its position on the Landis and Koch (1977) benchmark scale.

We will further dive deep into the causes of the interrater disagreement by detecting and categorizing various types of conflicts between elements used for annotation at the level of single data items.

5.3 Interrater Agreement and "Competing" Elements

In order to answer Research Question **2** – can we detect and categorize various types of conflicts between the existing ontology entities and the proposed elements via the interrater agreement scores? – we compiled for each data item a list of existing ontology entities and/or the proposed pattern types used to annotate this item. Then we calculated percent agreement and sorted the results in decreasing order. Percentages for the three most frequent choices for each data item are presented in Table 5.

The table shows that, for the majority of data items (76.78%, or 46/56), the agreement on the 1^{st} most frequent choice is $>$78%[7]. We interpret the percent agreement $>$20% for the 2^{nd} most frequent choice as a "competing" alternative.

Table 5. Percentages for the three most frequent choices for each data item. Proposed elements are colored dark grey, existing ontology elements – light grey.

ID	1,%	2,%	3,%	ID	1,%	2,%	3,%	ID	1,%	2,%	3,%	ID	1,%	2,%	3,%
1.0	86.96	8.7	4.35	**2.8**	82.61	4.35	4.35	**4.6**	100	0	0	**5.13**	82.61	13.04	4.35
1.1	82.61	8.7	4.35	**2.9**	86.96	4.35	4.35	**5.0**	95.65	4.35	0	**6.0**	95.65	4.35	0
1.2	95.65	4.35	0	**3.0**	91.3	4.35	4.35	**5.1**	100	0	0	**6.1**	78.26	17.39	4.35
1.3	95.65	4.35	0	**3.1**	91.3	4.35	4.35	**5.2**	95.65	4.35	0	**6.2**	95.65	4.35	0
1.4	95.65	4.35	0	**3.2**	91.3	4.35	4.35	**5.3**	86.96	13.04	0	**6.3**	86.96	8.7	4.35
2.0	65.22	21.74	8.7	**3.3**	91.3	4.35	4.35	**5.4**	86.96	8.7	4.35	**6.4**	86.96	8.7	4.35
2.1	91.3	4.35	4.35	**3.4**	86.96	8.7	4.35	**5.5**	86.96	8.7	4.35	**6.5**	34.78	34.78	21.74
2.1	100	0	0	**3.5**	91.3	4.35	4.35	**5.6**	82.61	13.04	4.35	**6.6**	100	0	0
2.2	86.96	8.7	4.35	**4.0**	100	0	0	**5.7**	100	0	0	**6.7**	43.48	30.43	17.39
2.3	86.96	8.7	4.35	**4.1**	95.65	4.35	0	**5.8**	52.17	21.74	21.74	**6.8**	69.57	13.04	13.04
2.4	78.26	8.7	4.35	**4.2**	86.96	8.7	4.35	**5.9**	65.22	21.74	4.35	**7.0***	56.52	17.39	8.7
2.5	95.65	4.35	0	**4.3**	100	0	0	**5.10**	65.22	30.43	4.35	**7.1***	65.22	17.39	13.04
2.6	95.65	4.35	0	**4.4**	91.3	8.7	0	**5.11**	43.48	26.09	17.39	**7.2**	78.26	13.04	4.35
2.7	86.96	8.7	4.35	**4.5**	65.22	26.09	4.35	**5.12**	52.17	47.83	0	**7.3**	69.57	21.74	4.35

In the results' distribution, one can find three types of alternatives:

[7] Relatively low agreement scores for the first chosen option in the data items 7.0 and 7.1 are most probably due to the equally probable conceptualizations. To annotate keys with values "busStopId" and "busStopName", participants could either first introduce a new class "BusStop", missing in the shared ontology, and then new properties where this class was a domain, or create new properties with the domains "basis:Bus" or "basis:BusLine".

- between two existing elements (e.g., "basis:incidentCause" and "schema:-description" in data item 2.0),
- between a pattern type and an existing term (e.g., an instance of the *Observable property* pattern and "schema:description" in 5.8; see also 4.5, 5.9, 5.11, 6.5),
- between two pattern types (e.g., instances of the *Observable property* and *Wildcard* patterns in 5.10; see also 5.12, 6.7, 7.3).

In 5 out of 6 cases, where the existing term was involved as the competing alternative, this term was "schema:description", a property with a very broad definition "A description of the item" and the "schema:Text" value type. In the systems where ontology extension by annotators is enabled, the presence of elements with such a general meaning hinders interoperability. The majority of participants still chose either a more specific property ("basis:incidentCause" in data item 2.0), or created properties with a specialized meaning (e.g. by instantiating the *Observable property* pattern in 5.8 to describe the availability status of a charging dispenser).

Another case where an existing ontology element is selected as an alternative to a pattern instance is a property "basis:carSpeed". It was used to annotate data item 4.5 with a key name "GPS Speed"; 65.22% created a new property with a specialized meaning by instantiating the *Observable property* pattern. Using the existing element is not wrong in principle. Nevertheless, reusing "basis:carSpeed" to annotate "GPS Speed" leads to ambiguity: two keys in Data sample 5 are described with the same property in the shared ontology. In this case, if the shared ontology is not extended, annotating the key with an existing entity creates semantic interoperability conflict.

The group where two pattern types were competing as the 1^{st} and the 2^{nd} most frequent choices, contains three combinations. Firstly, the *Wildcard* pattern is used along with the *Observable property* pattern (items 5.10 and 5.12). In both data items, they introduce properties related to sensor measurements: one, indicating whether a charging plug has quick charge support, and the other – a maximum current of a charging dispenser. These properties represent static attributes of a plug and a dispenser rather than the values measured by sensors. This shows that such boundary cases can cause inconsistencies while introducing properties with similar meaning. Discussing this difference in the introductory materials will foster more uniform modeling.

In data item 6.7, in order to introduce the property to annotate the key "country", nested in the object "city" (along with the key "name"), 43.48% of the participants chose the *Spatial* pattern and 30.43% – the *Wildcard* pattern. In the description of Data sample 6, a city is defined as a location for which the weather forecast is provided. Here the "country" attribute is related to spatial modeling, but it differs from the postal address element used in the address specification for buildings. Both modeling solutions are possible and, if any of them is preferable, this case should also be discussed in the introductory materials.

Data items 7.0–7.3 go beyond prototypical sensor measurement contexts. They model the output of a service which estimates the waiting time before the next bus arrives at a bus stop. Even though the agreement in choosing the *Temporal* pattern to annotate the key "busTimeToArrival" was 69.57%, 21.74% of the participants modeled it as an *Observable property* instance. The latter modeling is also appropriate as this value is calculated based on a schedule or a position of the bus, which differs from the modeling of the resulting timestamp of a "sosa:Observation". This example clearly points to the limitations of the existing pattern set (see further discussion in Section 5.4) and the need to reconsider the modeling principles of the shared ontology (if similar datasets are modeled in the data cataloguing or integration solution).

To sum up, the presence of “competing” alternatives indicates the need to revise the shared ontology and a corresponding pattern set (in the case where very dissimilar patterns are being confused) or to discuss specific modeling scenarios during the training/introductory phase. The interrater agreement scores serve as reliable indicators of possible collisions.

5.4 Interrater Agreement and Semantic Interoperability Conflicts

In response to Research Question **3** – can we detect and categorize various types of possible semantic interoperability conflicts while using a single template? – we will successively examine the modeling choices in all types of patterns used to introduce new elements. Agreement metrics will reveal the aspects of semantic interoperability related to the possible semantic interoperability conflicts (after J. Park, [27]).

In general, we observe (see Tables 6–9) a high level of agreement in specifying properties’ domains and ranges in all patterns except for the instances of *Temporal* pattern. Agreement on the domain indicates that in most cases ontology extension will not introduce schema-isomorphism conflicts. Agreement on the range is related to the value type and possible data-representation conflicts.

Table 6. Percent agreement in the modeling choices made in the *Observable property* pattern instances.

ID	schema:	schema:	meta:property-	meta:unit
	domainIncludes	rangeIncludes	Semantics	
3.4	95.0; 5.0	100.0	80.0; 10.0; 5.0 × 2	75.0; 15.0; 5.0 × 2
4.2	100.0	100.0	35.0; 30.0 × 2; 5.0	95.0; 5.0
4.4	100.0	100.0	66.67; 23.81; 4.76 × 2	57.14; 23.81; 9.52; 4.76 × 2
4.5	100.0	100.0	93.33; 6.67	93.33; 6.67
5.8	100.0	91.67; 8.33	66.67; 25.0; 8.33	100.0

In Table 6, we present percent agreement on each of the chosen pattern fillers in the ***Observable property*** pattern instances, where this pattern was selected by the majority of the participants to introduce a new property. A hundred percent agreement means that there was only one value; all other cases show the split between various options, sorted in decreasing order[8]. High variability in the specification of the units of measurement (e.g., for the data sample 4.4, the options were: “KilopascalAbsolute”, “Kilopascal”, “Hectopascal”, “Pascal”) was somewhat surprising because all units were explicitly mentioned in the descriptions of the data sources. Even though this situation is less likely in the real setting (data providers must know their data, in the experiment it could be a matter of attention), the data annotation system should consider mechanisms to check the unit of measurement to avoid possible data-unit conflicts.

We also explored tendencies in how the participants specified the higher level property semantics. The options here were not mutually exclusive in terms of splitting the meaning continuum (see “meta:Quantity”, “meta:Amount”, “meta:Mass”, etc.). The results suggest that higher agreement is achieved for the options mentioned in the label of

[8] By using multiplication, e.g. “5.0 × 2”, we indicate that there we two options, each chosen by 5.0% of the participants.

an annotated element or in the data source description: “GPS Speed” – “meta:Speed” (93.33%, data sample 4.5), “Intake Pressure” – “meta:Pressure” (66.67%, data sample 4.4); or in the label of the related properties: “meta:Concentration” was chosen by 80% of the participants after annotating several keys with properties labeled “basis:noxConcentration”, “basis:coConcentration”, etc.

In the results dataset we did not observe confusion between the concepts of the highest level of abstraction: “meta:Quantity” and “meta:Quality”. As an example, in data item 5.8, the semantics of a property introduced to annotate the availability status of a charging dispenser was marked as “meta:Quality” in 66.67% cases; the second most selected option was “meta:NoneOfTheListed” (25%).

The results show that in the contexts focused on achieving semantics interoperability, the options should be mutually exclusive. In the future experiments, we will split the set of options into two lists and display them in the separate user interface forms. The first one – {“meta:Quantity”, “meta:Quality”} – will contain options with most general distinction that enable alignment with top-level ontologies. The second will be oriented towards capturing the subsumption relationships (e.g., properties which indicate speed values “carSpeed” and “GPSSpeed” will be the siblings of one superclass “meta:Speed”).

The instances of the ***Temporal*** pattern (see Table 7) demonstrated the greatest level of variability. We start with the data items 5.13 and 6.8. First and foremost, the idea of specifying a property which is observed by a sensor at a particular timestamp (the association which is essential to reconstruct the *sosa:Observation* context) was not understood by the participants. In the introductory video, we gave a short explanation of this relationship, which was obviously not sufficient to make reliable choices. This may also be due to the unspecific prompt “Which property is related to the proposed one?” used as a label for the options list. The majority of replies were split between the options: “meta:AllSensorMeasurementsInASample” and “meta:Other”. The former seems more appropriate because in both datasets there are multiple observable properties and only one time indicator.

Table 7. Percent agreement in the *Temporal* pattern instances.

id	schema: domainIncludes	schema: rangeIncludes	meta:temporal Semantics	meta:service Semantics	meta:observable Property
5.13	84.21; 15.79	89.47; 5.26; 5.26	100.0	100.0	26.32 × 2; 10.53; 5.26 × 7
6.8	81.25; 12.50; 6.25	56.25; 31.25; 12.50	100.0	62.50; 37.50	43.75 × 2; 6.25 × 2
7.3	43.75; 18.75 × 2; 12.5; 6.25	56.25; 18.75; 12.50 × 2	81.25; 18.75	93.75; 6.25	43.75; 37.50; 12.50

Another cause of disagreement was the usage of infrequent datatypes for the data values (see the column “schema:rangeIncludes” in Table 7). For the Unix epoch time in data item 6.8, the split was 56.56% – 31.25% – 12.5% for the options “meta:EpochTime”, “meta:DateTime”, “meta:Date”. Similar distribution was obtained for the duration value (“P0DT0H1M12S”) in data item 7.3. This variability is a potential cause of the data-representation semantic interoperability conflicts and should be carefully handled.

The options that specified temporal semantics [1] were rated most consistently in all samples: the replies contained at most 2 options, with the agreement ranging from 81.25% to 100.0% on the most frequent choice.

Service semantics for the weather forecast (data item 6.8), despite the dataset description ("for a particular time in the future"), was specified as measured, not predicted by the 37.5% of the participants. This distinction obviously needs some elaboration in the future training or assistance materials.

We finally discuss a more complex temporal property in the last data item 7.3. The value of this property presents the output of a service which calculates time remaining until the bus arrives at the bus stop. Here the modeling depends on the conceptualization of the situation: whether this attribute is seen as characterizing an arriving bus or a bus stop or even a bus line. The study participants chose "basis:Bus" class as the domain of the property in 43.75% cases, 31.25% of the participants introduced a new class for a bus stop und used it as a domain; another options were – "meta:BusLine" (18.75%) and "proposed:TimeToArrival" (6.25%). This is one of the examples where we have a clear indicator of 1) the multiple modeling decisions and 2) the limitations of the core temporal modeling in the *sosa:Observation* pattern, which aims at capturing the time at which the phenomenon took place. Note that more general temporal modeling (columns "meta:temporalSemantics" and "meta:serviceSemantics") shows a higher level of agreement even for very dissimilar properties.

Table 8. Percent agreement in the *Spatial property* pattern instance.

id	schema: domainIncludes	schema: rangeIncludes	meta:spatial- Semantics	meta:spatial- Structure
6.7	100.0	100.0	90.0; 10.0	100.0

The instances of other pattern types – *Spatial* (Table 8) and *Wildcard* (Table 9) – demonstrate a very strong agreement between participants. The *Wildcard* pattern was introduced to collect properties not covered by the core template set. The agreement is not surprising – this pattern simply allows the establishment of relationships between classes, or between a class and a datatype. Nevertheless, these relationships can be further specialized into a new pattern. For example, the *Wildcard* instances in the current result set showed the need of a pattern to express the "part-of" relation (e.g., "basis:Dispenser" *hasPlug* in 5.9). We also need to differentiate a static characteristic of an object (e.g., "basis:Plug" *hasQuickChargeSupport* in 5.10) and the changing state measured by a sensor (see properties like "basis:plugAvailabilityStatus").

Table 9. Percent agreement in the *Wildcard* pattern instances.

id	schema: domainIncludes	schema: rangeIncludes	id	schema: domainIncludes	schema: rangeIncludes
5.9	93.33; 6.67	86.67; 13.33	5.11	100.0	100.0
5.10	93.33; 6.67	93.33; 6.67	5.12	83.33; 16.67	91.67; 8.33

In this section, we showed how various types of semantic interoperability conflicts can result from the modeling choices made in the process of using templates. The agreement scores obtained for the options clearly indicate these problematic cases, where the unified behavior should be enforced either by training or by automated checks and suggestions. Also these scores point in many cases to limitations of the shared ontology.

6 Conclusions

This work outlines an approach to explore various aspects of semantic interoperability, related to the annotation of IoT data, by computing the interrater agreement scores (percent agreement and the Fleiss' kappa) on a set of annotation tasks. To the best of our knowledge, it is the first experiment where annotation of IoT data with the elements of one shared ontology is examined in the situation when the rater not only uses an ontology (or a set of ontologies), but extends it if needed elements are missing.

We show the role of the interrater reliability metrics in pointing out the "competing" elements which might be introduced as the result of the ontology extension process. We also describe how interrater agreement scores can be used to detect potential semantic interoperability conflicts, in particular, data-representation, data-unit and schema-isomorphism conflicts.

We suggest methods to model proposed elements with various levels of granularity and to investigate the impact of modeling choices on overall agreement in a complex annotation-extension task. In the experiment with 23 domain experts, we obtained initial estimates for the overall agreement and scores on single items. We will further use these estimates to improve the implemented prototype of the annotation system tailored to the IoT data.

We believe that our approach can be applied outside the IoT domain as well as generalized from the data cataloguing and/or data integration solutions to more complex tasks and environments. In general, it contributes to a much wider topic – how meaning is created in the community of users, and thus, to realizing the vision of the Semantic Web: "...the relations allow communication and collaboration even when the commonality of concept has not (yet) led to a commonality of terms" [5]. Exploring other contexts where shared meaning is created and agreed upon in a distributed manner will be one of the directions of our future research.

References

1. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. In: Weld, D.S. and de Kleer, J. (eds.) Readings in Qualitative Reasoning About Physical Systems. pp. 361–372 Morgan Kaufmann (1990)
2. Andročec, D. et al.: Using Semantic Web for Internet of Things Interoperability: A Systematic Review. Int. J. Semant. Web Inf. Syst. 14, 4, 147–171 (2018)
3. Agarwal, R. et al.: Unified IoT Ontology to Enable Interoperability and Federation of Testbeds. IEEE 3rd World Forum on Internet of Things (WF-IoT) 70–75 (2016)
4. Bermudez-Edo, M. et al. IoT-Lite: a lightweight semantic model for the Internet of Things. Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, IEEE, 90–97 (2016)
5. Berners-Lee, Tim, James Hendler, and Ora Lassila. The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American 284.5 (2001): 3
6. Broering, A. et al.: Enabling IoT ecosystems through platform interoperability. IEEE software 34.1, 54–61 (2017)
7. Chui, M. et al.: Applying artificial intelligence for social good. Discussion Paper. November 2018. https://www.mckinsey.com/featured-insights/artificial-intelligence/applying-artificial-intelligence-for-social-good . Last accessed 10 May 2019

8. Cohen, J.: A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement. 20, 1, 37–46 (1960)
9. Dietze, H. et al.: TermGenie: A web-application for pattern-based ontology class generation. J Biomed Semantics. 5, (2014)
10. Dumitrache, A. et al. Empirical methodology for crowdsourcing ground truth. Semantic Web 1, (2018)
11. Egana Aranguren, M. et al.: Transforming the Axiomisation of Ontologies: The Ontology Pre-Processor Language. Nature Procedings. (2009)
12. Fleiss, J.L.: Measuring nominal scale agreement among many raters. Psychological Bulletin. 76, 5, 378–382 (1971)
13. Forssell, H. et al.: Reasonable Macros for Ontology Construction and Maintenance. In: CEUR Workshop Proceedings. Vol. 1879. Technical University of Aachen (2017)
14. Gamer M. et al: irr: Various Coefficients of Interrater Reliability and Agreement. R package version 0.84.1. https://CRAN.R-project.org/package=irr (2019)
15. Gangemi, A. et al.: A Pattern-based Ontology for the Internet of Things. In: WOP@ISWC. (2017)
16. Ganzha, M. et al.: Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective. Journal of Network and Computer Applications. **81**, 111–124 (2017)
17. Gonzalez-Usach, R. et al.: Interoperability in IoT. In Handbook of Research on Big Data and the IoT. IGI Global, (2019)
18. Gil, Y. et al.: A Controlled Crowdsourcing Approach for Practical Ontology Extensions and Metadata Annotations. In: d'Amato, C. et al. (eds.) The Semantic Web – ISWC 2017. pp. 231–246 Springer International Publishing, Cham (2017)
19. Janowicz, K. et al.: SOSA: A lightweight ontology for sensors, observations, samples, and actuators. Journal of Web Semantics. (2018)
20. Jayaraman, P. P. et al.: The Schema Editor of OpenIoT for Semantic Sensor Networks. In: Joint Proceedings of the 1st Joint International Workshop on Semantic Sensor Networks and Terra Cognita (SSN-TC 2015) and the 4th International Workshop on Ordering and Reasoning (OrdRing 2015). Bethlehem, Pennsylvania, US (2015)
21. Jupp, S. et al.: Webulous and the Webulous Google Add-On–a web service and application for ontology building from templates. J Biomed Semantics. 7, 17 (2016)
22. Karlsson, D. et al.: Semantic Krippendorff's α for measuring inter-rater agreement in SNOMED CT coding studies. In MIE, (2014)
23. Khan, M. T.: Involving domain experts in ontology construction: A template based approach. In: Extended Semantic Web Conference. Springer, Berlin, Heidelberg, (2012)
24. Kottner, J. et al.: Guidelines for Reporting Reliability and Agreement Studies (GRRAS) were proposed. Journal of Clinical Epidemiology. 64, 1, 96–106 (2011)
25. Landis, J.R., Koch, G.G.: The Measurement of Observer Agreement for Categorical Data. Biometrics. 33, 1, 159 (1977)
26. O'Connor, M.J. et al.: Mapping Master: A Flexible Approach for Mapping Spreadsheets to OWL. In: Patel-Schneider, P.F. et al. (eds.) Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part II. pp. 194?208 Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
27. Park, J.: Information systems interoperability: What lies beneath? ACM Transactions on Information Systems. **22**(4), 595–632 (2004)
28. Soldatos, J. et al.: OpenIoT: Open source Internet-of-Things in the cloud. In Interoperability and open-source solutions for the Internet of Things Springer, Cham, 13–25 (2015)
29. Stevens, R. et al.: Measuring expert performance at manually classifying domain entities under upper ontology classes. Journal of Web Semantics. (2018)
30. Vetere, G., Lenzerini, M.: Models for semantic interoperability in service-oriented architectures. IBM Systems Journal **44**(4), 887–903 (2005)
31. Wickham, H.: Reshaping Data with the reshape Package. Journal of Statistical Software, 21(12), 1–20 (2007)