I2oTology - Tracking-Oriented Ontology

Levy Marlon Souza Santiago[1], **Jauberth Weyll Abijaude**[1], **Péricles de Lima Sobreira**[1], **Fabíola Gonçalves Pereira Greve**[1]

[1]Departamento de Ciências Exatas e Tecnológicas
Universidade Estadual de Santa Cruz (UESC) – Ilhéus, BA – Brazil

levyssantiago@gmail.com,

{jauberth,plsobreira}@uesc.br,

fabiola@ufba.br

Abstract. *To join the Internet of Things (IoT) and Ontology concepts today it is becoming a good strategy to save sensors and Smart Objects (SO) information using all the semantic capabilities and ontology inferences to improve and give some intelligence at the information manipulation, IoT-Lite and SSN (Semantic Sensor Network) are examples of ontologies for IoT. This paper presents the I2oTology, which is a tracking-oriented ontology. The I2oTology purpose is to presents a semantic aimed at tracking smart objects based on some IoT-Lite classes. It was made a simple test with this ontology but there is some classes, properties and situations to be tested and also to know how much the ontology is right, these topics will be considered at the future work.*

1. Introduction

Internet of Things (IoT) adopts novel processing, communication architecture, smart technologies and management strategies to seamlessly integrate a large number of smart objects with the Internet [Li et al. 2018]. Service-oriented architectures (SOA) and methodologies have been widely adopted and studied in distributed systems, well before the emergence of IoT. However, due to huge number of entities and large diverse service pool in IoT, the trend has shifted towards using more lightweight services. Traditional SOAP-based services have been gradually replaced by RESTful services and APIs are now the new players in this field. APIs are easier to define, invoke, share, and monitor compared to other service definition methods [Khodadadi and Sinnott 2017].

One of the most highlighting features of the Internet of Things domain is the heterogeneity of the information. One method to accomplish this interoperability is through the usage of semantic-based technologies to annotate all the information shared by the platforms [Agarwal et al. 2016]. The ontology alignment allows organizations to model their own knowledge without having to stick to a specific standard [Gil et al. 2018].

Ontology is a representation vocabulary, often specialized to some domain or subject matter. More precisely, it is not the vocabulary as such that qualifies as an ontology, but the conceptualizations that the terms in the vocabulary are intended to capture [Chandrasekaran et al. 1999]. The main benefit of having an ontology for a specific domain is for confederacy and dissemination of knowledge about the domain and connecting with other domains [Keat and Shahrir 2017]. The OWL (Web Ontology Language) is a ontology language which facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics [McGuinness et al. 2004].

The IoT-Lite is a lightweight semantic model for IoT proposed by [Bermudez-Edo et al. 2016]. It's an instantiation of the Semantic Sensor Network (SSN) ontology [Compton et al. 2012]. The intent of IoT-Lite is not to be a full ontology for the IoT, it was built to be a core lightweight ontology that allows relatively fast annotation and processing time.

Hermes Widget IoT, [Veiga et al. 2017], is a component that extends the Hermes context management system [Sene Júnior et al. 2014] representation layer allowing it to handle information obtained from any sensor with a web endpoint, the Hermes Widget IoT uses the semantics of IoT-oriented ontologies such as IoT-Lite and SSN, it allows any context provider object, for example, a sensor, to be located, used, and have its corresponding context information represented and made available for querying through the Internet, the **geo:Point** class [Brickley 2006] allows the system define a geographic location of these context provider objects. The I2otology also integrates this class but is still under tests, but the idea is to locate the SO's as the Hermes Widget does.

A semantic Industrial Internet of Things (IIoT) architecture is proposed by [Pease et al. 2017]. Between the architecture layers, there is an ontology called IIoT ontology which uses RDF and OWL for knowledge engineering. This ontology contains the tracking ontology class IndoorTrackingDevices used to link device type to service, for service functionality discovery. Also there is an inventory of assets which can be located in real time, so the system "knows" when an asset is moving. The I2otology follows the same concept, it can give some information to the system that is using it about which SO is moving, who is moving it and where is its actual location. But the idea is to use RFID portals in rooms, in this way if the Smart Object get out of a room, the system will "know" that this object is moving because its RFID tag will be read.

The remainder of this paper is organized as follows. The next section is an explanation about the I2oTology, the merging with IoT-Lite and a brief inference example. The Section 3 describes the I2oTology implementation in a web system. Conclusions and future work direction in Section 4.

2. The I2oTology

The I2oTology (Figure 1) is an extention of IoT-Lite. There's some IoT-Lite classes that wasn't used, and there's some classes added (Table 1) to give some others capabilities which will be explained as follow. The proposed ontology presents a semantics aimed at tracking smart objects making use of IoT-Lite location classes, as well as adding the possibility of a reasoner to discover through this ontology if a specific object can be in a given room with a certain person.

This ontology has been applied within a specific context (an university), but in the question of tracking it can be said that it can be applied to similar contexts. The I2oTology was tested (next section) with a web system which is still being developed to be used at the UESC University.

Table 1. I2oTology's classes explanations

Institution	This class represents institutions where this ontology is applied, in this case, the UESC University.
Place	This is a superclass that represents a general place, it can be a classroom, a laboratory and others.
Laboratory	This class represents laboratory rooms and for this reason, it's subclass of Place.
ClassRoom	This one represents classrooms and as well as the **Laboratory** class, this is a subclass of **Place**.
AllowedPlace	This is an equivalent class[1]. When an smart object individual "*canBeIn*" some place (e.g. **Laboratory**) and has the relation *hasLocation* to some kind of that place (e.g. individual "lab_18"), so the reasoner will infer that this object is in the allowed place.
Person	This class represents people. It's used to define the person that is moving some object.
Teacher	This class represents teachers and is subclass of **Person**.
Functionary	A class that represents institution functionaries. Its also a subclass of **Person**.
Student	Students representation, also a subclass of **Person**.
AllowedPerson	This class follows the same idea of AllowedPerson. It's an equivalent class that represents if the object is been moved (using the *canBeMovedBy* and *isMovedBy* properties) by an allowed person.
RfidSensor	Represents Rfid readers. It is a subclass of **SensingDevice**
RfidSensorCharacteristic	This class represents some rfid characteristics, for example, it's Antenna(s) and this classes can be related by the *hasAntenna* property.
Antenna	A class that represents the rfid antenna(s).
InfraredSensor	This class represents infrared sensors that for this project it can be used to warn to turn on/off some rfid sensor where some presence is detected. For that case the **InfraredSensor** class *controls* the **RfidSensor**.
TurnOnInfraredSensor	This is an equivalent class that is "activated" when the data property *detectedPresence* is *true*.
Material	This one represents object materials. This class have two subclasses **Dangerous** or **Simple**, it's just to mark the material type that can influence at the moving time.
Dangerous	This class represents dangerous materials (e.g. chemical).
Simple	This class represents simple materials (e.g. plastic).
AllowedMaterial	As the **AllowedPerson** and **AllowedPlace**, this is an equivalent class that represents if the object is of a simple material (using *ofMaterial* property).

2.1. Merging I2oTology and Iot-Lite

The Figure 1 shows the merge between I2oTology and IoT-Lite. Since I2oTology is a tracking-oriented ontology, IoT-Lite's **Object** class does not apply to it because it characterizes the family of objects that aren't part of a **Device**, such as a desk and a chair. Objects like these in I2oTology receive an RFID code and are therefore treated as **TagDevice**. For the location implementation strategy, the **Laboratory** and **ClassRoom** classes were created, these are subclasses of **Place** and because of this, it was not necessary to use the Deployment class, which in the end, was replicating information that those classes store. The *hasLocation* and *hasAttribute* properties were adapted for the project, so they didn't take IoT-Lite in the nomenclature once they were changed. The **Teacher**, **Functionary** and **Student** classes were added to represent and classify people, these are subclasses of **Person** class and participate in the traking process to define who is moving the device. The **AllowedPlace**, **AllowedPerson AllowedMaterial**, and **TurnOnRfidSensor** classes are equivalent classes used to "trigger some event". It is from them that the ontology will give suggestions and confirmations to the System used to test it.

The **Material** class is used only to define the material that the device is made of, it may be **Dangerous** or **Simple** types. As subclasses of the **SensingDevice**, **RfidSensor** and **InfraredSensor** have been added, which are the types of sensors that this System communicates. An **RfidSensor** can have multiple antennas, this is why the **Antenna**

[1] Equivalent classes are necessary conditions. When an individual follows exactly this conditions, it'll be inferred that this is an individual of this class.

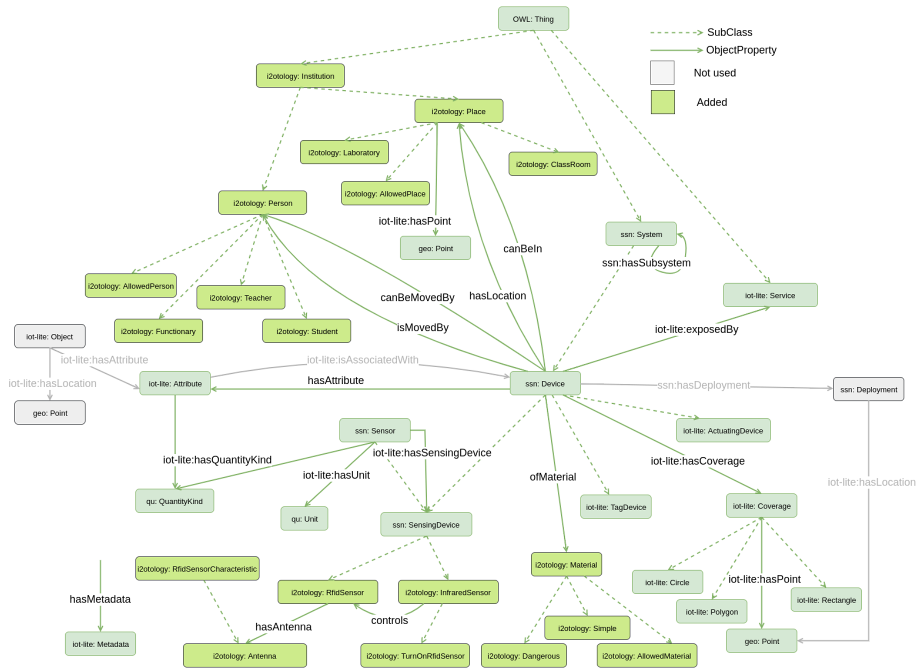

Figure 1. The I2oTology

class of **RfidSensorCharacteristic** was added.

2.2. How it works

Here is a simple example showing how is the inference of I2oTology (ilustration at Figure 2). Assuming the following registered individuals: obj1, func1 (is a **Functionary**), stud1 (is a **Student**), plastic (is a **Simple** material) and room1 (is a **Laboratory**). The obj1 has a *deviceTag* (a data property added) that is “f3h532w”, it *canBeIn* some **Laboratory**, *canBeMovedBy* some **Functionary** and is *ofMaterial* plastic. This object *isMovedBy* stud1 and right now *hasLocation* room1. If the reasoner is started, there will be the following inferences:

- obj1 is a **TagDevice** because it has a *deviceTag*;
- obj1 is individual of **AllowedMaterial** because it’s of a simple material;
- obj1 is individual of **AllowedPlace** because it *canBeIn* some **Laboratory** and room1 is a **Laboratory**.

The obj1 isn’t individual of **AllowedPerson** because it *canBeMovedBy* some **Functionary** and stud1 is a **Student**. The ontology and it’s inferences in this context “answer” to the system (explained in next section) the SO status. The status information can show, for example, with who is the SO? Where is it right now? Can this person move this object? Can this object be in this place? With this information, the system can act when something wrong occurs.

3. Implementing the ontology

As said, the ontology presented was partially implemented in a web system to be tested. The project aims to implement a tracking system to improve the object verifications at the University. The system uses RESTful for server communication. To implement the ontology, it was created an ontology service layer to build a system-ontology communication.

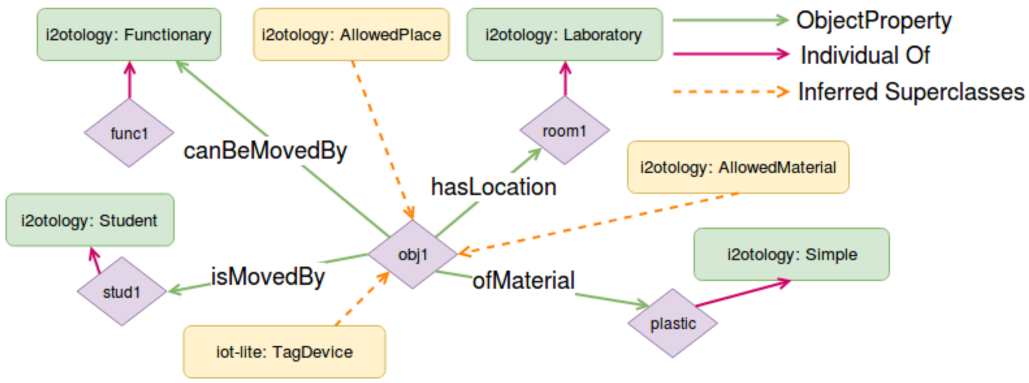

Figure 2. I2oTology Inference Example

To develop the ontology API to work with the I2oTology, it was used the Jena Framework [Jena 2007] for building Semantic Web and Linked Data applications.

To make a simple test with the I2oTology, some SO's were registered in both system's database and ontology. The web system has a page to insert some RFID tags read by a RFID reader. The user can access this page by some palm computer which is connected to RFID reader, read the tags and insert in the text area and press "send". After the user press "send", the system will verify if this object is registered in the database and then will send all the read tags to the ontology service (also with the actual location where all tags were read) to make all needed verifications and inferences. After some seconds, the ontology give the system some results, as well as if the SO is in the right place with the right person and the system prints each Smart Object result to the user. The system idea is to monitor each movement and activity of each object, knowing at all the time where it is, with who it is and if it can be with that person in this place, and to act in any way every time something wrong happens.

4. Conclusion and Future Work

This paper presented a tracking-oriented ontology, the I2oTology, which aims to be implemented in UESC University and in the question of tracking, it can be said that it can be applied to similar contexts. This ontology was partially tested with a system which is still being developed. This test consisted to evaluate the ontology response about which room is an Smart Object, who is moving it and if it can be with this person at this place. There are others classes, properties and situations to be tested with the I2oTology and this is some of the future work topics. Some of next steps is to implement a way to use the **geo:Point** class to track the exact place where the SO's are located (with latitude, longitude and altitude information), to integrate the system registry page with the ontology to create the SO in the database and in the ontology maintaining the data consistency and to know how much the ontology is right so with this results it'll be possible to be sure about the ontology certain.

References

Agarwal, R., Fernandez, D. G., Elsaleh, T., Gyrard, A., Lanza, J., Sanchez, L., Georgantas, N., and Issarny, V. (2016). Unified iot ontology to enable interoperability and federation of testbeds. In *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on*, pages 70–75. IEEE.

Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., and Taylor, K. (2016). Iot-lite: a lightweight semantic model for the internet of things. In *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences*, pages 90–97. IEEE.

Brickley, D. (2006). Basic geo (wgs84 lat/long) vocabulary. *Documento informal escrito en colaboración*.

Chandrasekaran, B., Josephson, J. R., and Benjamins, V. R. (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems and their applications*, 14(1):20–26.

Compton, M., Barnaghi, P., Bermudez, L., GarcíA-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al. (2012). The ssn ontology of the w3c semantic sensor network incubator group. *Web semantics: science, services and agents on the World Wide Web*, 17:25–32.

Gil, J. M., Montes, J. F. A., Alba, E., and Aldana-Montes, J. (2018). Optimizing ontology alignments by using genetic algorithms.

Jena, A. (2007). semantic web framework for java.

Keat, N. W. and Shahrir, M. S. (2017). Ontology for customer centric digital services and analytics. In *AIP Conference Proceedings*, volume 1905, page 040024. AIP Publishing.

Khodadadi, F. and Sinnott, R. O. (2017). A semantic-aware framework for service definition and discovery in the internet of things using coap. *Procedia Computer Science*, 113:146–153.

Li, J., Yan, Q., and Chang, V. (2018). Internet of things: Security and privacy in a connected world.

McGuinness, D. L., Van Harmelen, F., et al. (2004). Owl web ontology language overview. *W3C recommendation*, 10(10):2004.

Pease, S. G., Conway, P. P., and West, A. A. (2017). Hybrid tof and rssi real-time semantic tracking with an adaptive industrial internet of things architecture. *Journal of Network and Computer Applications*, 99:98–109.

Sene Júnior, I. G., Bulcão-Neto, R. d. F., et al. (2014). Anatomy of a semantic context interpreter with real-time events notification support. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*, pages 159–162. ACM.

Veiga, E. F., Arruda, M. F., Neto, J. A. B., and Bulcão-Neto, R. d. F. (2017). An ontology-based representation service of context information for the internet of things. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*, pages 301–308. ACM.