An ontology for the automated deployment of applications in heterogeneous IoT environments[1]

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Konstantinos Kotis[a, *], Artem Katasonov[a]
[a]*VTT Technical Research Centre of Finland*

Abstract. In the near future, Internet of Things should integrate an extremely large amount of heterogeneous entities. To tackle heterogeneity, these entities will need to be consistently and formally represented and managed (registered, aligned, composed and queried) trough suitable abstraction technologies. Two distinct types of these entities are a) sensing/actuating devices that observe some features of interest or act on some other entities (call it 'smart entities'), and b) applications that utilize the data sensed from or sent to the smart entities (call it 'control entities'). The aim of this paper is to present an ontology as the key technology for the abstraction of these entities, towards supporting the automated deployment of control entities in settings where smart entities have been already deployed. In specific, the paper presents the use of this technology to support the following required distinct tasks of this deployment process: a) the semantic registration of heterogeneous IoT entities, b) the alignment of IoT entities' metadata and use of these alignments for the matchmaking between smart and control entities, and c) the alignment of the semantics of the data of the messages that are exchanged between these IoT entities during device-to-application communication. The paper presents a use case scenario and the formal definitions of the main concepts and properties of the proposed ontology. Furthermore, the paper presents a proof-of-concept-based ontology evaluation approach.

Keywords: ontology, Internet of Things, semantic interoperability, ontology alignment, semantic matchmaking

[1] This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. This Programme is supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.
[*]Corresponding author. E-mail: Ext-Konstantinos.Kotis@vtt.fi

1. Introduction

Internet of Things (IoT) enables a global connectivity between the real world and a virtual world of entities (observed subjects/objects, sensing/actuating/identity/embedded devices, software applications/services), connects things, not only places or people, and brings real-time machine-published data/information to the users of the existing Web.

The 'things' on the IoT are various physical entities that are of some interest to humans, e.g. a heater to control, a package to track, an industrial machine to monitor, the air temperature in a room to measure, the motion in a room to detect. Depending on the nature of these 'things', different technologies for connecting them to the IoT are used: a) identity devices (e.g. RFID tags or barcodes), b) sensing and actuating devices (e.g. temperature and other sensors, cameras for cars' register plate recognition, and actuators like remotely-controlled door locks or window blind controls), and c) embedded electronics (e.g. industrial machinery, home electronics, smart phones and wearable devices that have embedded parts e.g. processors, data storages, sensors and actuators).

Due to the large diversity of millions of devices, IoT requires interoperability at multiple levels. On the hardware side, such problems have to be addressed as handling a capability mismatch between traditional Internet hosts and small devices, as well as handling widely differing communication and processing capabilities in different devices. In the interface between the device and network domains, IoT gateways will provide a common interface towards many heterogeneous devices and networks. Some IoT devices, e.g. home electronic appliances will be likely connected directly to the Internet without such middle-boxes. For the communication between heterogeneous IoT devices and applications towards the automated deployment of latter in environments where the former have been already deployed, there is currently a gap of a facilitator technology (interface). Third-party applications that are developed without being aware of devices' data models (descriptions), but being generic in the sense of being capable of running on various IoT device sets, will not be possible to be automatically deployed. Such inability is caused by the semantic gap between heterogeneous data of both types of entities i.e. devices and applications. Such a gap can be bridged by aligning the meaning of data that both types of entities may 'carry' and use these alignments for their semantic matchmaking.

Based on the above ascertainments, IoT entities need to be consistently, explicitly and formally represented and managed (registered, aligned, composed, and discovered) through suitable abstraction technologies i.e. ontologies. Such a representation and management capability will enable their seamless integration in different application domains of IoT, such as smart home, ambient assisted living, transportation, etc., in a way that deployment of third-party generic applications in non-expert end-users' IoT settings will be performed automatically, with minimum involvement of both non-experts and experts.

In this paper we focus on the use of semantic technologies, specifically ontologies, for the automated deployment of IoT entities, supporting the following three distinct tasks: a) the semantic registration of IoT entities, b) the alignment of IoT entities' metadata and use of these alignments for their matchmaking, and c) the alignment of the semantics of the data of the messages that are exchanged between these IoT entities during device-to-application communication. We view the deployment of third-party generic applications in such settings as an automated process that hides from end-users the complexity of tasks such as ontology alignment and semantic matchmaking of IoT entities. These complex tasks are not expected to be performed by non-expert end-users but rather they should be supported by an expert third-party service provider e.g. an IoT semantic interoperability as a service (SIaaS) provider that acts as a mediator between IoT application providers and consumers, placing the ontology alignment and semantic matchmaking tasks of the deployment process of IoT entities in a mediation layer of the IoT stack.

In this paper, we consider ontologies as a key technology to solve the problem of automating the deployment of applications in heterogeneous IoT environments, allowing any IoT entity to unambiguously convey the meaning of data/information they 'carry'. The aim of the presented ontology as an abstraction technology is to hide heterogeneity of IoT entities, acting as a mediator between IoT application providers and consumers, and to support their semantic matchmaking. Acting as a mediator, the ontology objective is to be used by the interested stakeholders independently as a registry for the semantic registration of IoT entities, i.e. by the IoT application providers/developers that will register their software and by the IoT application consumers/users and IoT device owners that will register their sensing/actuating/embedded devices and the associations of these devices with the observed subjects/objects. The ontology (as a semantic registry) must be engineered in an efficient and effective way so that the alignment and matchmaking of IoT entities (by appropriate software tools) will be facilitated.

The work reported in this paper is particularly motivated by our vision of an open and interoperable

IoT, where at least the following four related requirements must be satisfied:

a) Ability to have gradually growing IoT environments, contrasted to the need to install and interconnect all IoT devices and software at once
b) Ability to interconnect IoT entities from different vendors
c) Ability of 3rd parties to develop software applications for IoT environments, contrasted to applications coming only from the devices' vendors.
d) Ability to develop applications that are generic in the sense of running on various IoT device sets (different vendors, same purpose), contrasted to developing applications for a very particular configurations of devices.

The paper is structured as follows: Section 2 describes motivation and related work. Section 3 describes the aim, objectives and scope of the developed ontology, based on an indicative scenario of its use. Section 4 describes the conceptualizations formalized and section 5 reports on the ontology evaluation approach. Finally, concluding remarks and future plans are discussed.

2. Motivation and Related work

The main motivation of this work is driven by the need to support the automation of the process of the deployment of applications in heterogeneous IoT settings, with the minimum human involvement at the end-users' side i.e. providers and consumers of IoT applications, and with minimum human involvement at the experts-side i.e. experts on ontology alignment and matchmaking of IoT entities. As already introduced, such a process can be realized through an ontology acting as a semantic registry and by a set of tools for the alignment and matchmaking of the semantically registered (in the ontology) IoT entities. The lack of a suitable ontology i.e. an ontology with the required aim, scope and objectives, as well as the complete definitions of classes and properties needed to represent the required knowledge, further motivated us towards devising the proposed IoT-ontology.

Based on the abovementioned motivations and also on the 'abstraction level challenge" of the Semantic Sensor Web [11] and from the future work directions of the W3C XG's final report on the Semantic Sensor Network (SSN) ontology [12], we identified the lack of a set of high-level abstractions of IoT entities any of the existing related work.

More specifically, we identified the lack of the notion of a smart entity (SE) which corresponds to an abstract representation of the association between a) sensing/actuating/embedded/identity devices, b) features of interest that they are observe, and c) software agents that are responsible for the entity's conceptualization (domain ontology) and for entity's functionality (provided as a service).

In addition, existing works do not represent applications as IoT entities, similarly to SEs. In our context, we refer to these entities as control entities (CE). This lack of conceptualization of both types of IoT entities prohibits the task of automated alignment and matchmaking and, consequently, the automated deployment of them in heterogeneous IoT environments.

Last but not least, most of the existing IoT or sensor-related ontologies represent IoT devices only partially e.g. sensing devices in SSN ontology. This need for representing richer information related to IoT entities and their properties conforms also with one of the challenges of Semantic Web research in terms of smart entities, reported by Sabou M. in 2010 [10] i.e. the representation of a variety of information regarding smart objects on the IoT.

In the work of Christophe et al in 2011 [2], authors present ongoing work towards an ontological framework for the representation and retrieval of connected smart objects in the Web of Things [3]. The notion of a virtual object is partially in alignment with the smart entity notion presented in this paper. Also, the aim of this related work for discovering similarities between connected objects is in alignment with the aim of our ontology. Having said that, the related ontologies were not available for reuse and quite limited information was provided in the related research paper regarding the devised conceptualizations. More importantly, this work does not reuse the SSN ontology which is the latest W3C representational effort to reach consensus on the description of sensed data and observations.

In the related work reported by Hachem et al, 2011 [13], authors state that have devised a global ontology for the IoT domain. The work emphasizes the representation of devices and also the composition and estimation of measurements. However, the work does not reuse generic and widely-agreed ontologies such as the DUL and SSN ontologies. Most importantly, this work does not represent the complete knowledge related to all types of IoT entities and to their alignment. Finally, they are not publicly available for reaching consensus and for reusing them.

Related work in SOFIA[2], SENSEI[3], SemSorGrid4Env[4], SM4All[5] and HYDRA[6] projects has

[2] http://www.sofia-project.eu/
[3] http://www.sensei-project.eu/
[4] http://www.semsorgrid4env.eu/

been focusing more on middleware frameworks rather than on the modeling of related knowledge, regarding sensors in smart spaces or federated sensor networks towards solving small or large-scale data integration problems. A latest effort within the SWISS-Experiment[7] project emphasizes the use of W3C XG SSN ontology in a large-scale federated sensor network for semantic data sensor search [4]. The SSN ontology is a domain-independent ontology that describes sensors and observations for use in sensor network and sensor web applications, by merging sensor-focused, observation-focused and system-focused views (considering and integrating most of the conceptualization efforts of the other related projects). The SSN ontology has been devised by taking into consideration the aforementioned related projects, so reusing it in our work was a key design decision. The aim and scope of the ontologies developed for these projects are a subset of the one presented in this paper since they focus only on sensing devices-related information and do not abstract into the higher-level notions of smart and control entities, towards supporting their alignment and matchmaking (for automating the deployment of applications in heterogeneous IoT environments).

An IoT conceptual model presented as part of the IoT-A project [1] defines abstractions of IoT entities as virtual entities but in a different way, i.e. a virtual entity is a model of a physical entity e.g. a human, a car or even a temperature sensor. Also, this ontology does not represent applications as IoT entities that can be semantically registered similarly to smart entities. Last but not least, the ontology fails to follow widely accepted and agreed ontology design patterns such as the one for modeling sensors and observations as the Stimulus-Sensor-Observation Ontology Design Pattern [10] and it does not reuse any existing high-level ontology e.g. DUL, or IoT/Sensors-Network-related ontology, e.g. SSN.

Concluding, the modeling and ontology creation effort presented in this paper does not attempt to compete against related ones but, in contrast, it strives towards reusing them where and when possible, in order to better support its aim, objectives and scope in the most efficient and effective way.

3. Ontology aim, objectives and scope

In this section we outline the aim, objectives and scope of the ontology. To better realize these, we present a scenario that accentuates the need for a semantic registry as a central point for the alignment and matchmaking tasks of the IoT entities' automated deployment process.

3.1 An IoT entities' matchmaking and deployment scenario

In our scenario we have used the domain of home automation, pointing on the need for an IoT semantic interoperability provider to act as a mediator between home automation/building management application providers and consumers. The actors of this scenario are shown in Table 1.

A.	**IoT application consumers and infrastructure owners**: buy IoT devices and applications for their house
B.	**IoT infrastructure and data providers:** sell sensors, actuators, gateway boxes for home automation, IoT data feeds from the Cloud
C.	**IoT application providers**: sell applications for the IoT home automation/building management domain
D.	**IoT semantic interoperability providers:** provide a platform (as a service) for the semantic registry and the alignment/matchmaking of IoT entities. Acts as a mediator for the deployment of IoT entities

Table 1. Real world physical entities: IoT entities' consumers and providers

In the same real world, other physical entities are also involved, entities that are related to the IoT infrastructure, data and applications, provided by different vendors. In Table 2 we outline examples of such physical entities, organized according to the actors that provide/consume them.

1.	**IoT application consumers and infrastructure owners** (e.g. Mary)
a.	House
b.	Alarm lights/lamp
c.	Air-condition
2.	**IoT infrastructure and IoT data providers**
a.	Vendor-A Temperature sensor
b.	Vendor-B Motion detection sensor
c.	Vendor-C Switch actuator
d.	Vendor-D Luminosity sensor
e.	Vendor-E Gateway box
3.	**IoT application providers** (e.g. building management company, offering security and energy saving applications/services)
a.	Providers's application for security monitoring
b.	Providers's application for energy monitoring

[5] http://www.sm4all-project.eu/
[6] http://www.hydramiddleware.eu
[7] http://www.swiss-experiment.ch

4.	IoT semantic interoperability providers
a.	Semantic registry
b.	Semantic interoperability software Platform (tools for IoT entities' ontology alignment, matchmaking, message translation, message transformation, etc)

Table 2. Real world physical entities: IoT infrastructure, data and applications/services

Mary is leaving home for a holiday and wants to manage the security and the energy consumption of her house while abroad. She contacts a building management (BM) service provider where she is informed about the availability, cost and deployment requirements of the company's related applications as a service (AaaS), i.e. a) a security service, and b) an energy saving service. The necessary infrastructure (equipment) for these services to be able to run is not provided by BM so Mary is instructed to buy them from a third-party company. The necessary sensor/actuator devices that Mary bought i.e. a temperature sensor, a motion detection sensor, a luminosity sensor and two switch actuators, are able to connect to the outside world i.e. to the Internet, via a gateway box that she also bought from a different vendor. As instructed by BM, Mary uses the Web interface of an IoT semantic interoperability provider in order to register the installed IoT equipment. Via a Web interface, the IoT semantic interoperability provider service allows Mary to simply describe (e.g. using a template-driven form) the installed infrastructure and provide information on a) their ID and address e.g. URI, b) what physical entities they are associated with (observe or act on), and c) the approximate labels for the required input and output data that will be exchanged with the BM IoT application/service provider. Using the same Web interface, the BM IoT application provider registers the applications that Mary bought for the security and energy saving of her house. At this point it must be noted that between the IoT infrastructure in Mary's house and the IoT semantic interoperability provider, an additional actor such as an IoT data provider (e.g. COSM) may be used to store the data coming from Mary's IoT personal infrastructure, i.e. sensor data. Also, it must be noted that the MB IoT application provider may have (or not) already associated some domain ontology with the input/output data required for the application to function (annotate the input/output parameters of the related service using the semantics of a domain ontology, in our case, a building automation/management ontology). In fact, domain ontologies may or may be not used from all actors during the automated deployment of heterogeneous IoT entities. For instance, Mary may register her own IoT smart entities by just providing some free-text labels to describe their input/output functionality. These labels could be also browsed by a Web-provided domain ontology for home automation, if available.

Both Mary and BM have the freedom to initiate the deployment process of the registered applications in Mary's IoT registered infrastructure, at any time after the registration of all involved IoT entities. By the execution of the deployment process, the IoT semantic interoperability provider service executes the alignment and matchmaking tasks for the input and output parameters of the applications (control entities) against the input and output parameters of the services related to Mary's IoT infrastructure (smart entities). These first tasks match the IoT entities in order to identify which smart entities that 'live' in Mary's house are appropriate for the applications/services (that Mary bought) to function. The matched entities are stored in the ontology appropriately (using subclass axioms). Right after this matchmaking and discovery stage, the deployment process initiates another task in order this time to align the data 'carried' in the communication messages that are exchanged between the already matched (from previous tasks) IoT entities. This latter alignment task is also required since application developers and IoT device vendors have the freedom to describe the input/output data of their IoT entities in different ways i.e. using different domain ontologies. These alignments are also stored in the ontology appropriately (using the corresponding alignment cell structures).

At the end of a successful alignment and matchmaking of IoT entities process, the IoT semantic interoperability provider service notifies Mary and BM that the deployment process is successfully ended and that the services are ready to run. The IoT semantic interoperability provider is responsible for resolving any conflicts in the matchmaking and alignment process during the deployment process, without involving Mary at any stage of this. The IoT semantic interoperability provider service is a (semi-)automated service that allows experts' involvement for the validation of the alignment and matchmaking tasks.

The provided scenario can be similarly used to describe any real-world application domain, by replacing the IoT application/service consumer and the IoT application/service provider according to the domain requirements. For instance, Mary can consume a new IoT application/service that is provided by a distant medical station that offers e-health monitoring services (environmental conditions and health status remote monitoring).

3.2 Ontology aim and scope

A formal and explicit representation of all types of IoT entities and their associations is required in order to serve as the semantic registry of the related to the above scenario real-world instances. Such an abstraction technology is needed in order to a) abstract (hide) the technological heterogeneity that derives from the vast amount of heterogeneous IoT entities, b) abstract (hide) the semantic heterogeneity that derives from the use of heterogeneous domain ontologies to semantically annotate the data of those IoT entities.

The abstract world of our scenario includes (but not limited to) the abstract entities presented in Table 3, organized according to the different levels of the physical world that we have identified. Examples from the scenario are provided for each level of the IoT abstract-world entities.

1.	**Smart entities (SE) abstracting Mary's physical world** a. A-SE_1 actuator-smart entity: switch actuator associated with a lamp b. A-SE_2 actuator-smart entity: switch actuator associated with an air-condition c. S-SE_1 sensor-smart entity: motion detection sensor associated with Mary's living room d. S-SE_2 sensor-smart entity: temperature sensor associated with Mary's living room) e. S-SE _3 sensor-smart entity: sunlight sensor associated with Mary's living room)
2.	**Control entities (CE) abstracting IoT application/service providers' physical world)** a. CE_1 control entity: abstraction of a simple smart home application for monitoring light in a room based on movement detection. Example application logic: IF motion is detected in Mary's living room THEN light switch actuator switches on the lamp. The CE is instantiated based on the application logic. b. CE_2 control entity: abstraction of a simple smart home application for monitoring room temperature via smart phone app. Example application logic: IF temperature is higher than 30 degrees Celsius in Mary's living room THEN air condition switch actuator acts on LG air condition, setting the power on. The CE is instantiated based on the application logic.

Table 3. Abstract world: IoT abstract entities and example instantiations from the scenario

3.3 Ontology objectives and high-level requirements

In this section we outline ontology's detailed objectives and high-level requirements that have drove its design and development. These objectives and requirements are aligned with the main goal of proposing an ontology as a semantic registry for the automated deployment of applications in heterogeneous IoT environments, and also aligned with the two-fold aim of the ontology, as abovementioned, i.e. a) to abstract (hide) the technological heterogeneity that derives from the vast amount of heterogeneous IoT entities, b) to abstract (hide) the semantic heterogeneity that derives from the use of heterogeneous domain ontologies to semantically annotate the data of those IoT entities.

The ontology's individual objectives are outlined in the following list:

1. support the semantic registration of heterogeneous IoT entities offered by infrastructure or application/service providers and used by application/service consumers
2. support the alignment of semantic descriptions of registered IoT entities (by representing the alignment input i.e. smart and control entities, and the alignment output i.e. aligned entities in the form of subclass axioms between them)
3. support the matchmaking of registered IoT entities based on the discovered alignments (by executing proper SPARQL queries on the updated with the alignments ontology)
4. support the alignment of communication messages between matched IoT entities (by representing the alignment input i.e. example IoT entities' communication message templates, and the alignment output i.e. aligned data descriptions of communication messages stored as instances of mapping cells of the related ontology alignments)

Based on the abovementioned objectives, the ontology's high-level requirements are outlined in the following paragraphs.

Requirement 1: *An IoT ontology must represent any IoT physical entity that needs to be registered, managed and involved in the deployment of IoT applications*. The IoT physical entities include (but not limited to) the following:

a) Hardware entities
   i. Sensing devices, actuating devices, identity devices, embedded devices
   ii. Physical entities/features of interest to be observed or to be changed (change their state by acting on them) e.g. home appliances, food products, transportation media, building spaces, etc.
b) Software entities

i. Applications
ii. Software agents
iii. Services

Requirement 2: *An IoT ontology must represent high-level IoT entities as abstractions of physical entities' associations*. The IoT high-level entities include the following:

c) Smart entities: an abstract representation of an association between
   i. Sensing or actuating or embedded or identity devices
   ii. Features of interest that they are related to
   iii. Software agents that are responsible for the entity's conceptualization (domain ontology) and for entity's functionality (provided as a service).

a) Control entities: an abstract representation of an association between
   i. Applications
   ii. Features of interest that they are related to
   iii. Software agents that are responsible for the entity's conceptualization (domain ontology) and for entity's functionality (provided as a service).

Requirement 3: *An IoT ontology must facilitate the representation of ontology alignment-related information, as this is captured in the two distinct alignment tasks:*

1. Alignments between IoT entities, based on the semantic descriptions of input/output parameters that application developers (for control entities) and end-users (for smart entities) provide
2. Alignments between semantic description of communication messages between matched IoT entities, based on the semantic description that application developers (for control entities) and device vendors (for sensing/actuating/identity/embedded devices) provide

3.4 Semantic Smart Gateway Framework and the IoT-ontology

In other lines of our related research [8, 9] we have argued that an IoT Semantic Smart Gateway Framework (IoT-SSGF) must provide a semantic registry for IoT entities and a set of tools to support ontology creation, ontology alignment, data message format transformation, etc. Peers (providers or consumers of IoT entities) must be able to automatically register new entities or retrieve already registered ones that match certain properties/criteria. Newly added heterogeneous entities, carrying their vendors' ontological descriptions (or other types of metadata) for describing properties of the devices or data they carry, must be coordinated by aligning their ontology definitions with the definitions of other related entities.

The IoT-ontology proposed in this paper supports the IoT-SSGF by representing different types of IoT entities that are fundamental parts of IoT domain, as well as by associating ontology definitions related to the alignment of ontologies (mapping cells of correspondent alignments between ontological classes and properties).

The IoT-ontology is only one part of the IoT-SSGF. The other part of the SSGF is a toolset that utilizes the ontology towards achieving the automated deployment of IoT applications. The description of this toolset, named Smart Proxy, is out of the scope of this ontology paper. A description of the toolset and preliminary evaluation of its modules can be found in a related recent publication [9]. Having said that, in the current paper we demonstrate the utilization of the ontology by this toolset as part of the ontology's evaluation approach.

4. Representing IoT entities and their alignments

This section presents the designed and implemented conceptualizations of the proposed IoT-ontology for the automated deployment of applications in heterogeneous IoT environments.

The proposed ontology is engineered using a layered approach. Conceptualizations related to sensing and observations were reused from the SSN ontology (*ssn* prefix for http://purl.oclc.org/NET/ssnx/ssn# namespace of the SSN layer) and high-level generic ones from the DUL ontology (*dul* prefix for http://www.loa-cnr.it/ontologies/DUL.owl# namespace of the DUL layer). On the limitation of existing ontologies to fully meet the requirements for the proposed IoT-ontology, the paper introduces the definition of new concepts and properties a) for the representation of higher-level entities of the IoT world i.e. smart entities and control entities, as associations between physical entities (sensing/actuating/embedded/identity devices, features of interest, software agents, applications and services), and b) for the representation of IoT entities' alignment information. For these new concepts/properties, we introduce two additional layers in the IoT-ontology, i.e. the IoT entities layer and the IoT entities' alignment layer. The *iot* prefix will be used for the IoT entities layer of the proposed IoT-ontology namespace, with base IRI: http://purl.org/IoT/iot#. Last but not least, the *align* prefix for the namespace

http://knowledgeweb.semanticweb.org/heterogeneity/alignment# will be used for the IoT entities' alignment layer.

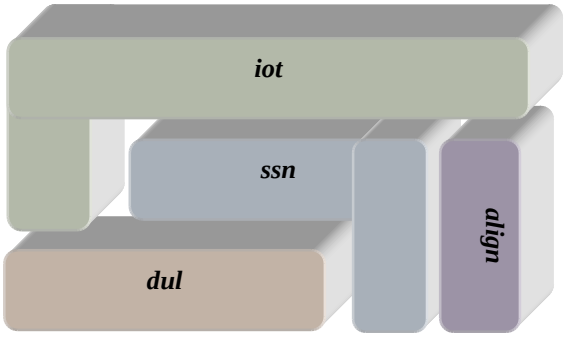

Fig. 1. IoT-ontology layers, indicating dependencies of different namespaces

In respect to the SSN ontology, our work can be viewed as an extension to it, by adding two new ontology layers (Figure 1) to support the requirements for an IoT-ontology as these are identified in this paper: the layer for representing IoT entities (IoT entities layer) and the layer for representing IoT entities' alignments (IoT entities alignment layer). Nevertheless, the proposed ontology extends the type of devices associated with different features of interest and goes beyond the notion of a sensing device i.e. adds the notions of actuating, identity and embedded devices. In addition, it introduces the notion of related services (actuating, observation and identity communication services) that are provided by associated software agents.

4.1 *IoT entities layer*

4.1.1 iot:IoT_Entity

The highest (top-level) concept of the IoT entities layer is the concept *iot:IoT_Entity* (Figure 2), specified as an Internet of Things entity that has a distinct, separate situation, existence or view and is consistent with ('satisfying') a description on a set of entities. It is a subclass of *ssn:Situation* (a social object that satisfies some description) and of *ssn:FeatureOfInterest* (a relation between an observation and the entity whose quality was observed) restriction to some *dul:PhysicalObject* (any object that has a proper space region). Furthermore, an IoT_entity is conceptualized by a software agent via an ontology, and this is defined as a property *dul:isConceptualizedBy* with restriction to some *iot:SoftwareAgent* that *dul:conceptualizes* some *iot:Ontology and iot:providesService some iot:Service*. The *iot:SmartEntity* and *iot:ControlEntity* are direct subclasses of *iot:IoT_Entity*.

Fig. 2. IoT_Entity definition in Protégé 4.2 notation showing the asserted and inherited axioms (as classes) in separate panes

Since, *ssn:Situation* and *dul:PhysicalObject* are defined in the imported namespaces (*ssn* and *dul* respectively), we do not present here their definition in detail. In this paper we present the definitions only for classes/properties defined in *iot* and *align* namespace.

4.1.2 iot:SmartEntity

An *iot*:*SmartEntity* (Figure 3) is an *iot:IoT_Entity* that represents the association between exactly one *dul:PhysicalObject* observed and some other objects (*dul:includesObject*) such as an *ssn:Sensor*, an *iot:Actuator*, an *iot:EmbeddedDevice*, *or an iot:Identifier*. The observation is represented using the *ssn:featureOfInterest* property restriction to *dul:PhysicalObject* i.e. a relation between an observation and the entity whose quality was observed e.g. in an observation of the weight of a person, the feature of interest is the person and the quality is weight. An *iot*:*SmartEntity* has an (inherited) property *dul:isConceptualizedBy* with restriction to some iot:SoftwareAgent that *dul:conceptualizes* some *iot:Ontology and iot:providesService some iot:Service*.

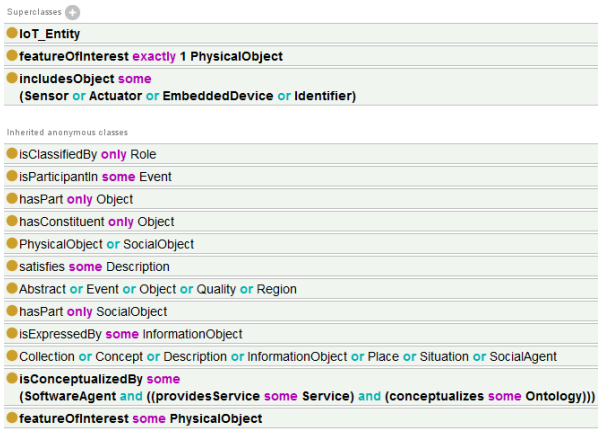

Fig. 3. SmartEntity definition in Protégé 4.2 notation

4.1.3 iot:ControlEntity

An *iot:ControlEntity* (Figure 4) is an *iot:IoT_Entity* that represents some control behavior/application logic. An *iot:ControlEntity* instance will to be matched to one or more instances of an *iot:SmartEntity* in order to achieve a specific application domain goal e.g. to monitor and adjust temperature in a room based on the sensed data. An *iot:ControlEntity* has an (inherited) property *dul:isConceptualizedBy* with restriction to some *iot:Application* (which is subclass of iot:SoftwareAgent) that *dul:conceptualizes* some *iot:Ontology and iot:providesService some iot:Service*.

Fig. 4. ControlEntity definition in Protégé 4.2 notation

Due to size limitation of this ontology paper, the rest classes/properties of the IoT entities layer can be viewed in N3/Turtle notation in the example presented in the evaluation section of this paper and also in the OWL implementation of the proposed ontology at https://dl.dropbox.com/u/1311500/IoT-Ontology/IoT-Ontology.zip and at http://purl.org/IoT/iot. A hierarchical view of the classes of all layers is also provided in the Appendix.

4.2 IoT entities' alignment level

As already stated, the key for true interoperability of IoT entities towards their automated deployment is to support their alignment and matchmaking of their data semantics (at deployment time). This must be done with minimum human involvement at expert's side i.e. at the SIaaS providers side, and with minimum human involvement at non-experts end-users' side i.e. at the IoT services/applications consuming point (e.g. at Mary's house). The goal here is to compute and store alignments in such a consistent and formal way that they could be utilized later on for the communication of the IoT entities (at runtime). The IoT-ontology considers the representation of ontology alignments in the way presented in the following paragraphs.

4.2.1 iot:OntologyAlignment

An *iot:OntologyAlignment* is a *dul:Description* for representing ontology alignments via *iot:aligns* restriction on exactly two instances of an *iot:Ontology*. It *iot:hasAlignmentCell* restriction on zero or more instances of an *iot:AlignmentCell*. IoT entities are linked to their ontology alignments via the *iot:Ontology* class of the *iot:aligns property restriction.*

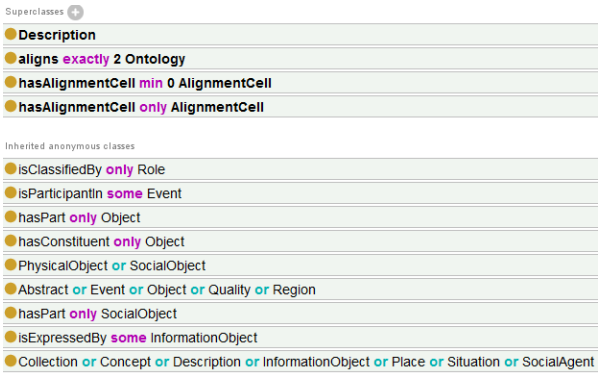

Fig. 5. OntologyAlignment definition in Protégé 4.2 notation

4.2.1 align:AlignmentCell

Based on the ODP initiative design pattern for alignments [6] and on the Alignment API [5], an *iot:OntologyAlignment* (Figure 6) has zero or more instances of an *iot:AlignmentCell*. An *iot:AlignmentCell* is a *dul:Description* and a *align:Cell*, representing the structure and type of an alignment cell between an *align:entity1* and an *align:entity2* (i.e. ontology classes or properties). It

also represents the correspondence (*align:relation)* that holds between them (i.e. equivalence, subsumption, etc), and the measurement (*align:measure)* of the confidence value (in the interval [0,1]) of this correspondence.

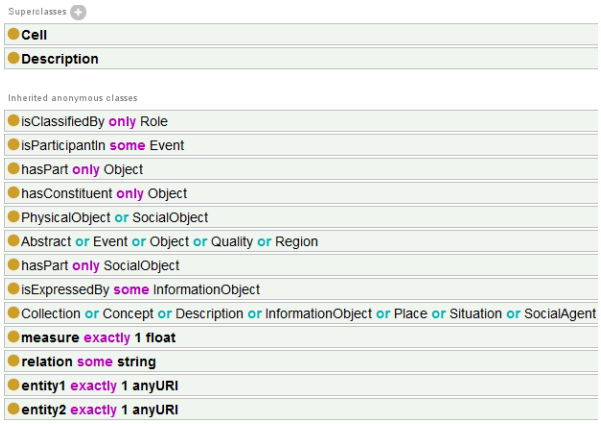

Fig. 6. AlignmentCell definition in Protégé 4.2 notation

4.3 Methodological issues

The engineering of the proposed IoT-ontology follows an evolutionary and open development process, towards the definition of a set of shared and community-agreed representational primitives with which to model the IoT domain of knowledge [14]. Based on the requirements gathered in the context of latest IoT-related research projects and motivated by key challenges reported in latest related research reports/papers, mostly from the SSN and IoT community, the main methodological decision is to reuse and extend existing conceptualizations and ODP patterns, and at the same time to improvise new ones towards supporting the aim and scope of the proposed ontology. Due to received (and expected) feedback from members of the SW and IoT community, a versioning strategy is also considered. Finally, the evaluation of each ontology version, in addition to the argumentation-based evaluation process, as proposed by HCOME OE methodology [7], is currently performed by experimentally using instantiated versions of the ontology with a proof-of-concept prototype of early IoT-SSGF implementations (Smart Proxy toolset).

5. Evaluation

We have used Protégé 4.2 Alpha and HermiT 1.3.6 reasoning engine to formalize conceptualizations and check inconsistencies of the proposed ontology. We have also used OpenRDF Sesame 2.6 as a semantic data store, for the instantiation and querying of example individuals. For the IoT-SSGF implementation, we have developed custom tools that support a) the transformation of JSON/XML/URI messages to OWL, b) the automated alignment and matchmaking of OWL specifications (concepts/properties), and c) the bi-directional translation of IoT entities' messages using pre-computed alignments.

The presented OWL version (v2.1) of the developed ontology reuses *ssn*, *dul* and *align* namespaces. A zip file with the related .owl files can be retrieved from https://dl.dropbox.com/u/1311500/IoT-Ontology/IoT-Ontology.zip and http://purl.org/IoT/iot.

In the N3/Turtle example use of the ontology provided below, for the sake of brevity, only the most relevant properties for IoT entities' registration are presented.

Let a physical object of interest, a lamp, be registered (by the end-user) in the repository as:

```
:Lamp a dul:DesignedArtifact, :LampType .
:LampType a owl:Class; rdfs:label "Light"@en .
```

The label "Light" was provided by a human (the end-user of IoT application and infrastructure owner, e.g. Mary) as a free-form annotation of the class of the object.

Let then an actuating device, the remote-controlled electrical socket, be registered as:

```
:Switch a iot:Actuator, iot:ActuatingDevice.
```

and the association of the lamp and the device (lamp is plugged into and thus controlled through the socket) is expressed as a smart lamp entity:

```
:SmartLamp a iot:SmartEntity;
        ssn:featureOfInterest :Lamp;
        dul:includesObject :Switch.
```

The service provided by :SmartLamp, the :LightSwitchService, is also registered in the repository but is omitted here.

Another smart entity, i.e. a smart room entity is described below. Its feature of interest is a room:

```
:E023 a iot:Room .
```

The sensing device associated with this physical entity is a motion detector:

```
:MotionDetector a ssn:Sensor, ssn:SensingDevice .
```

The smart room entity is registered in the ontology as follows:

```
:SmartRoom a iot:SmartEntity;
        ssn:featureOfInterest :E023;
        dul:includesObject :MotionDetector;
        dul:isConceptualizedBy [
                a iot:SoftwareAgent;
                iot:providesService :DetectionService
        ] .
```

The service provided by this smart entity through its associated software agent is registered as follows (only the output's description is given here for simplicity):

```
:DetectionService a iot:Service;
    ssn:hasOutput [
        a iot:InformMessage;
        dul:expresses [ a : MotionDetectorSignalType ];
        dul:isRealizedBy [
            a iot:CommunicationMessage;
            rdfs:label "MotionDetectorSignal";
            iot:hasTemplate "{'list': [ { 'timeStamp':
1333450241.736228, 'signal': 'PropertiesChanged', 'data': {
'MotionDetected': true }, 'IDeviceId': 23 } ], 'until':
1333450241.741899, 'tobj': 'signals' }"
        ]
    ];
    iot:hasAccessAddress
      <http://192.168.50.1/api/signals/listen?ids=23> .

:MotionDetectorSignalType a owl:Class;
    rdfs:subClassOf ssn:Property;
    rdfs:label "Motion"@en.
```

The label "Motion" was also provided by a human (the end-user of IoT application and infrastructure owner, e.g. Mary) as a free-form annotation of the type of information expressed by the output of the above service.

Now let us assume that a generic application has been developed, implementing the function "switch a light when a movement is detected in the room". This application will be registered in the ontology (by the IoT service provider and application developer) as an application that provides some light service and conceptualizes a control entity:

```
:Control a iot:ControlEntity;
        dul:isConceptualizedBy :Application .
:Application a iot:Application;
        iot:providesService :LightService .
```

The instantiation of the specific service that the IoT service provider (application developer) provides is described as follows:

```
:LightService a iot:Service;
    ssn:hasOutput [
        a iot:QueryMessage;
        dul:isRealizedBy [
            a iot:CommunicationMessage;
            rdfs:label "Monitor";
            iot:hasTemplate "token=X"
        ];
        iot:hasTarget "?entity ssn:featureOfInterest [a
iot:Room]; dul:isConceptualizedBy [iot:providesService
?this]. ?this ssn:hasOutput [dul:expresses [ a
<http://www.vtt.fi/IoT/test/app1#HasMovement> ] ]"
    ];
    ssn:hasInput [
        a iot:InformMessage;
        dul:isRealizedBy [
            a iot:CommunicationMessage;
            rdfs:label "MovementDetectorResponse";
            iot:hasTemplate
"<event><type>movement</type><value>false</value></eve
nt>"
        ]
    ];
    ssn:hasOutput [
        a iot:CommandMessage;
        dul:isRealizedBy [
            a iot:CommunicationMessage;
            rdfs:label "Switch";
            iot:hasTemplate
"<command><type>switch</type><value>1</value><token>
X</token></command>"
        ];
        iot:hasTarget "?entity ssn:featureOfInterest [a
<http://www.vtt.fi/IoT/test/app1#LightingDevice>];
dul:isConceptualizedBy [iot:providesService ?this]"
    ].
```

Additionally, the custom classes used in *iot:hasTarget* SPARQL patterns have to be defined along with the application registration:

```
app1:HasMovement a owl:Class;
        rdfs:label "HasMovement"@en.
app1:LightingDevice a owl:Class;
        rdfs:label "LightingDevice"@en.
```

The Smart Proxy toolset utilizes the abovementioned registrations using the following steps in order:

1. Apply the ontology alignment task on the registrations as a whole, aligning classes

:LampType (label "Light") and :MotionDetectorSignalType (label "Motion") with app1:LightingDevice and app1:HasMovement, correspondingly. Such alignments are stored in the ontology using subclass axioms, e.g. :LampType rdfs:subClassOf app1:LightingDevice.

2. Perform the matchmaking between the registered application and the smart entities by executing patterns of the iot:hasTarget-related SPARQL queries. For "Monitor" query, the pattern uses IoT-ontology to define the application's requirements for a sensing service that expresses a "hasMovement" property of a room. For the "Switch" command, the pattern express the application's requirement for an actuating service provided by a smart entity associated with a "LightingDevice" feature of interest. Although these patterns are simple for presentation issues, they can be made as complex as needed within the borders of expressivity power of the IoT-ontology. The alignment performed in step 1 will ensure that the results are properly returned. Variable "?this" will contain in each case the service sought for.
3. Process the message data format examples given with iot:hasTemplate for both the smart entities' services and the control entity's application. This is done by first generating an OWL-based ontology representation for given JSON or XML and then applying the ontology alignment toolset on those. As a result, "MotionDetected" attribute in :DetectionService's JSON output will get mapped with the "value" XML tag in the application :LightService's MovementDetectorResponse input. Similarly, all other patterns defined with the application and smart entity registrations are aligned. Such alignments are stored into the repository simply as values of the properties of iot:AlignmentCell class (align layer).

6. Concluding Remarks

This paper has presented the engineering of an IoT-ontology to support several tasks of the automated deployment process of applications in heterogeneous IoT environments. The paper presented a scenario to support the need for a semantic registry for the IoT entities in order to a) abstract (hide) the technological heterogeneity that derives from the vast amount of heterogeneous IoT entities, b) abstract (hide) the semantic heterogeneity that derives from the use of heterogeneous domain ontologies to semantically annotate the data of those IoT entities.

The work reported in this paper is particularly motivated by our vision of an open and interoperable IoT, where the ultimate goal is to allow any IoT application/service provider to develop applications that are generic in the sense of running on various IoT device sets (different vendors, same purpose), contrasted to developing applications for very particular configurations or types of devices. The paper has conjectured that existing ontologies do not fully cover the aim, objectives and scope of the proposed ontology, and furthermore, they partially integrate the required conceptualizations. Based on that, the engineered IoT-ontology reuses widely accepted and agreed conceptualizations from SSN and DUL ontologies, integrating open ontology design patterns from the ODP initiative, and at the same time it introduces new ones.

In addition to the description of the conceptualizations of the IoT-ontology, the paper presents an evaluation of the latest version of the devised ontology, based on a motivating scenario, using an OWL implementation and a proof-of-concept Java implementation of the IoT-SSGF (Smart Proxy toolset for the registration, alignment, matchmaking and querying of IoT entities).

Future work includes a more elaborated version of the IoT-ontology, a modularization effort, and verification of its alignment with SSN and DUL. It also includes the investigation of further engineering requirements of the IoT-ontology in order to support (if needed) the representation of compound IoT entities. Last but not least, a full documentation and detailed report of the proposed ontology will be included in the near future plans.

Acknowledgements

Authors acknowledge that part of the work presented in this paper has been driven by initial requirements that have been specified in the following

projects: Internet of Things (TEKES Finnish project) and iCore (FP7 project). We also thank several anonymous and eponymous contributors during the argumentation-based ontology evaluation process as well as the reviewers that provided valuable comments in the first round of reviews of this paper.

References

[1] A. Nettsträter, Internet of Things Architecture (IoT-A) project, Deliverable D1.3 - Architectural Reference Model for the IoT, Jul, 2012.

[2] B. Christophe, V. Verdot, V. Toubiana, Searching the 'Web of Things', In: Fifth IEEE International Conference on Semantic Computing (ICSC), 2011, pp. 308 – 315

[3] D. Guinard, V. Trifa, E. Wilde, Architecting a Mashable Open World Wide Web of Things, Technical Report No. 663, Department of Computer Science, ETH Zurich, February 2010.

[4] J. Calbimonte, H. Jeung, O. Corcho and K. Aberer, Semantic Sensor Data Search in a Large-Scale Federated Sensor Network, In: Semantic Sensor Network Workshop, ISWC 2011, Bonn, 2011

[5] J. David, J. Euzenat, F. Scharffe, C. Trojahn dos Santos, The Alignment API 4.0, Semantic Web - Interoperability, Usability, Applicability, 2(1):3-10, 2011, IOS Press

[6] K. Janowicz and M. Compton, The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology, In: The 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010), 7-11 November 2010, Shanghai,China, 2010

[7]K. Kotis, A. Papasalouros, G. A. Vouros, N. Pappas, and K. Zoumpatianos, Enhancing the Collective Knowledge for the Engineering of Ontologies in Open and Socially Constructed Learning Spaces, Journal of Universal Computer Science, vol. 17, issue 12, pp. 1710--1742, 08/2011.

[8]Kotis, K., Katasonov, A., Leino, J. 2012. Aligning Smart and Control Entities in IoT. In: S. Balandin et al. (Eds.): 12th International Conference, NEW2AN 2012 and 5th Conference on Internet of Things and Smart Spaces, ruSMART 2012, LNCS 7469, Springer, pp. 39-50

[9]Kotis, K., and Katasonov, A. 2012. Semantic Interoperability on the Web of Things: The Smart Gateway Framework. In Proceedings of the Sixth International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2012), Palermo, 2012

[10]M. Sabou, Smart Objects: Challenges for Semantic Web Research, Semantic Web - Interoperability, Usability, Applicability, 1(1):1-5, 2010, IOS Press

[11]O. Corcho and R. Garcia-Castro, Five Challenges for the Semantic Sensor Web, International journal of Semantic Web - Interoperability, Usability, Applicability, 1(1):121-125, 2010, IOS Press.

[12]P. Barnaghi, M. Compton, O. Corcho, Semantic Sensor Network XG Final Report, W3C Incubator Group Report 28 June 2011, Semantic Sensor Network Incubator Group, Laurent Lefort, Cory Henson, Kerry Taylor (eds), http://www.w3.org/2005/Incubator/ssn/XGR-ssn/.

[13]S. Hachem, T. Teixeira, V. Issarny, Ontologies for the Internet of Things, In: ACM/IFIP/USENIX 12th International Middleware Conference, 2011

[14]T. Gruber, Ontology, in the Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag, 2009.

APPENDIX A

Overview of the IoT-ontology class hierarchy – some classes from imported namespaces are also depicted to support proper visualization of the hierarchy. Iot and align namespace classes are double-circled and on the right side of the vertical line that devices the depicted hierarchy.