Trust Semantics in IoT Entities' Deployment

Konstantinos Kotis
Department of Digital Systems, University of Piraeus
Karaoli & Dimitriou 80
Piraeus Greece
kotis@aegean.gr

George A. Vouros
Department of Digital Systems, University of Piraeus
Karaoli & Dimitriou 80
Piraeus Greece
georgev@unipi.gr

ABSTRACT
Semantics for the IoT domain have been already introduced in several semantic interoperability approaches, towards supporting the (semi-)automated deployment of generic IoT applications in environments where heterogeneous third-party IoT devices are deployed. Depending on the level of interoperability, an application may have to 'decide' which IoT devices in that environment are trustworthy for ensuring deployment and for avoiding low quality of services provided. In the open IoT, where a large number of generic applications and third-party devices co-exist, the need to ensure trustworthy deployment of system components is highly important. In this paper we present a simple and extensible modeling approach towards supporting this IoT task. Using fuzzy semantics as an enabler of trust in IoT, we demonstrate how trust can be seamlessly integrated in IoT ontologies. Doing so, these can serve as a secure *selection* key to an IoT application for selecting, among the available entities, the one(s) that the application should trust for its effective deployment in a specific context.

Keywords
IoT trust; semantic interoperability; trust semantics; IoT deployment

1. INTRODUCTION
Entities (applications, devices, sensors, humans, gateways, etc.) that 'live' in open, distributed and heterogeneous IoT environments 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 enables their seamless integration in different application domains, such as smart home, ambient assisted living, transportation, etc., in a way that deployment of generic applications and third-party devices in non-expert end-users' IoT settings is performed (semi-)automatically, with minimum human involvement.

Depending on the level of interoperability in the IoT environment and the ability of its dynamic expansion, an IoT entity may have to 'decide' which other entities in that environment are trustworthy, and then map its individual security policies with those trustworthy IoT entities in order to avoid critical 'misunderstandings'. This decision requires the ability of a generic application or third-party device to distinguish an entity as a *trustworthy* one. In the open and distributed IoT, where a large number of generic applications and third-party devices will be registered in different available registries, the need to ensure deployment of heterogeneous IoT entities highly important. To achieve this, there is a need to extend existing semantic interoperability approaches with *trust semantics*. When seamlessly integrated in IoT ontologies, trust can serve as a *secure selection key* of a generic IoT application/service to choose, among the available third-party registered devices, the one(s) that should be used for its effective deployment in a specific environment/context.

On the other hand, data in IoT is provided by different data sources. Trustworthiness of sources can be represented by *trust semantics* that describe quality of data and trust-related attributes for their providers and the sources themselves. Semantics can play an important role for defining trust and reliability attributes [1]. In addition, the high level of heterogeneity in IoT is expected to magnify security threats during the interaction of humans, machines, and robots, in any combination [2]. Furthermore, multiple heterogeneous IoT entities located in different contexts exchange information with each other, and this complicates the design and deployment of efficient, interoperable and scalable security mechanisms. The size and heterogeneity of the IoT affects its trust [3, 4]: a) trust in the interactions between entities, and b) trust in the system, from the users' perspective.

There are open trust-related issues that the state of the art in IoT needs to address, such as managing trust without the existence of central authorities, and those issues require clear and simple semantics towards solving interoperability as a first step (before going into 'deeper' security issues). Trust management mechanisms have been widely studied in various research fields. However, current IoT research has not comprehensively investigated how to manage trust in IoT in a holistic manner [4]. Seamless integration and cooperation of trust management mechanisms for achieving a holistic trust management in IoT is needed. The definition of a distributed and dynamic approach suitable for the scalable and open IoT context is still missing [2]. The introduction of a well-defined trust negotiation language supporting the semantic interoperability of IoT context, is still an open IoT-trust management issue [2].

The aim of this paper is to semantically enable trust in distributed and open IoT in order to ensure the deployment of heterogeneous IoT entities, without the existence of central trust authorities. By providing a degree of trustworthiness between heterogeneous IoT entities at the higher level of abstraction it is possible to ensure that the deployment of heterogeneous entities in the open IoT will be performed in a way that selecting entities with the higher trust value will be supported.

Towards this aim, the paper presents a simple but effective approach with the following contributions:

a) Propose a novel method for easy extension of any IoT ontology, introducing simple and extensible semantics related to trust between IoT entities

b) Reuse *trust semantics* from existing trust models/ontologies [8][9]
c) Define *trust semantics* using the existing framework of *FuzzyOwl2,* a fuzzy extension of OWL 2.

The paper is structured as follows: section 2 provides related work on trust modeling for IoT, and section 3 briefly discusses the main background concepts of semantic interoperability in IoT, fuzzy semantics and trust. Section 4 presents the proposed modeling approach along with a working scenario and section 5 concludes the paper, discussing further issues and work in-progress.

2. RELATED WORK

In [8], an ontology of trust is defined, specifying two types of trust, trust in belief (trust based on an agent believing in what another agent believes) and trust in performance (trust based on believing that another agent will perform an activity correctly). The 'trustor' (object property) is the agent performing the trusting and the 'trustee' is the agent that is being trusted. A 'trust degree' is a number between zero and one that signifies the degree to which the trustor trusts the trustee. A working ontology is available at http://ontology.eil.utoronto.ca/trust.owl. This related work focuses on the transitivity of trust in social networks.

In [9] authors contact an extensive survey and classify thirteen computational trust models by trust decision input factors. Their analysis is used to propose a new ontology for trust to facilitate interaction between business systems, focusing its utilization in digital business. A working ontology file in the related paper's corresponding URL (http://www.cs.helsinki.fi/u/viljanen/trust.owl) was not accessible (broken link) during the preparation of our paper.

In [10] authors introduce an ontology for trust representation that extends (with recent trust theories) another existing model [11] that focuses on the computational part of trust, rather than on social and agent aspects. Although the presented model is an updated extension of other efforts towards modeling trust, it focuses on the specific issue of trusting (Web) data. A working ontology file is not available (at least, not mentioned in the related paper).

In our work, we have studied the related trust ontologies and reused their common semantics as well as those that can contribute to our objectives. To the best of our knowledge, there isn't any related effort of integrating *trust semantics* in IoT ontologies towards supporting interoperability, aiming to ensure automated deployment of IoT entities in specific IoT environments where a centralized trust authority is not present.

3. PRELEMINARIES

3.1 Semantic Interoperability in IoT

In previous work [6], authors focus on the use of semantic technologies for the automated deployment of heterogeneous and distributed 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 messages' data that are exchanged between these IoT entities during device-to-application communication.

They've considered 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 IoT 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 (Figure 1), by the IoT application providers/developers that will register their software and by the IoT application. The IoT-ontology proposed [6] is mainly reusing the Semantic Sensor Network (SSN) ontology and the upper ontology DUL, supporting the IoT-SSGF framework by representing different types of IoT entities that are fundamental parts of the IoT domain. 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 real-world entities (Figure 1).

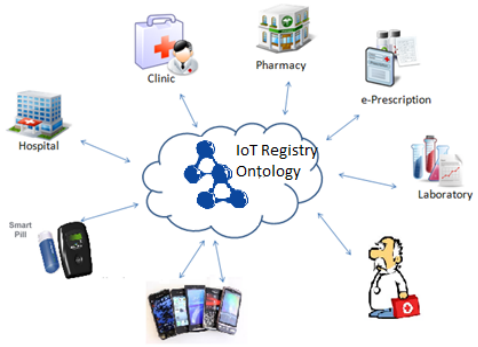

Fig. 1. IoT ontology as a semantic registry of entities

A short example of using the ontology is provided here (borrowed from this research line [6]), demonstrating the registration of a Smart Room and a Smart Lamp entity in a smart room scenario (a lamp is switched on if motion is detected). The reuse of SSN and DUL ontologies and the use of new IoT concepts can be observed in the Turtle-syntax examples provided below. Details on the full definitions and further examples are provided in [6].

"Smart Room" example description:

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

"Smart Lamp" example description:

```
:Lamp a dul:DesignedArtifact, :LampType .
:LampType a owl:Class; rdfs:label "Light"@en .
:Switch a iot:Actuator, iot:ActuatingDevice.
:SmartLamp a iot:SmartEntity;
        ssn:featureOfInterest :Lamp;
        dul:includesObject :Switch.
```

Let us now 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 IoT ontology (by the IoT service provider and application developer) as an application that provides some light service and conceptualizes a control entity. The instantiation of the specific service that the IoT service provider (application developer) provides are described in detail in [6], however here we provide the definition of a control entity that provides a light service:

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

As it is depicted in Figure 2, the execution of a 3rd party generic application developed for home security is utilizing a set of devices, communicating with them via a gateway box, and a message

translator that utilizes the uncovered and aligned (at deployment time) semantics of both parts, i.e. the application part and the devices part. But how these semantics have been computed? This may have been computed at the deployment time using a set of sample messages.

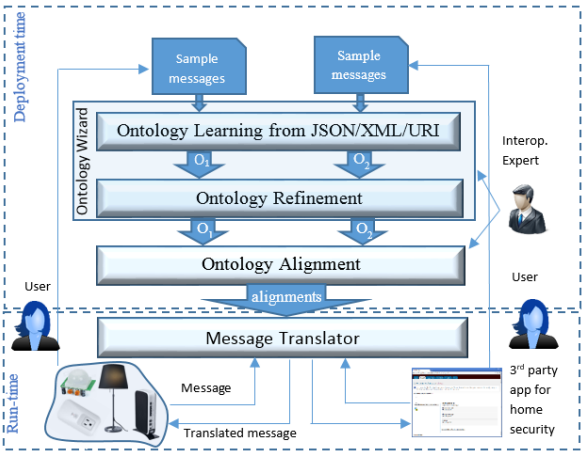

Fig. 2. The Smart Proxy architecture instantiation for 'smart room' scenario

An 'ontology wizard' (Figure 2) component is responsible for transforming messages that are exchanged between IoT devices and applications, (e.g. in JSON or XML or URI format) to ontological definitions of OWL classes and properties, as well as to refine those using some heuristic rules (e.g. to handle structural issues). The two sets of ontology definitions, one set for the device and one for the application, are then processed by an 'ontology alignment' component in order to obtain their similarities and compute alignments between them. These alignments (computed at the deployment time) are then used by the 'message translator' component at run-time for a bi-directional translation of messages.

The work presented in this paper is based on the abovementioned previous work of Smart Proxy and Semantic Smart Gateways Framework (SSGF) [6] and reuses the proposed IoT ontology as an example of extending IoT semantics with *trust semantics*. By providing a degree of trustworthiness between heterogeneous IoT entities at the higher level of abstraction it is possible to ensure that the deployment of applications in the open IoT will be performed by selecting devices with the higher trust values.

3.2 Fuzzy Semantics

The introduction of trust in terms of confidence values in the interval of [0, 1] for relations between concepts and properties has been extensively explored in *fuzzy ontologies* topic. The degree that an IoT entity is related with another IoT entity through a particular semantic relation (e.g. App x trusts Device y for its Functionality z in the Context w) can be used in open environments to avoid unauthorized/untrustworthy communication between 'foreign' entities as well as to play the role of a secure selection key for automated deployment, in environments with no central trust authority. In our work we use fuzzy ontologies as a *semantic enabler* for trust in IoT.

In this article we consider a fuzzy axioms of the form $\varphi \geq \alpha$ or $\varphi \leq \beta$, where φ is a fuzzy proposition and $\alpha, \beta \in [0, 1]$. This imposes that the degree of truth of φ is *at least* α (resp. *at most* β). For example, the proposition '*x is a reliable temperature sensor* ≥ 0.9' says that we have a rather reliable temperature sensor (the degree of truth of x being a reliable temperature sensor is at least 0.9). A (binary) *fuzzy relation* R over two countable classical sets X and Y is a function $R: X \times Y \rightarrow [0, 1]$. In this work we use the fuzzy extension of OWL 2, fuzzyOwl2 [7]. The use of annotation properties in this formalism allows *a)* to use current OWL 2 editors for fuzzy ontology representation, and *b)* OWL 2 reasoners to discard the fuzzy part of a fuzzy ontology, producing almost the same results as if it would not exist. Fuzzy OWL 2 assumes three alphabets of symbols, for *fuzzy concepts*, *fuzzy roles* and *individuals*. In fuzzy OWL 2, fuzzy concepts denote fuzzy sets of individuals and fuzzy roles denote fuzzy binary relations.

3.3 Trust

An attempt to produce a general definition and conceptual analysis of trust (and of the related idea of trustworthiness) has been recently made by O'Hara [12]. According to this report, '*trust is an attitude that one takes to the trustworthiness of another; in turn, the other's trustworthiness is a property that they have*'. Trustworthiness can be expressed as a quadruple:

$$\text{Tw} <\text{Y, Z, R(A), C}>$$

Y and Z are entities, R is a representation of behavior aimed at an audience A, and C is a context. This states that Y is trustworthy, assuming that there is some context for Y's trustworthiness. The context C is some type of relevant restriction of the circumstances in which Y is claimed to be willing, able and motivated to conform to R. In our current work, R represents the behavior of 'being reliable' in a specified context and task. Furthermore, if Y is trustworthy in all (or most) specific contexts where she has a duty, or is claimed, to be trustworthy, then it is generally trustworthy.

Trust is an attitude toward the trustworthiness of an entity for achieving specific goals or performing in particular ways in specified contexts. If X trusts Y, then X has a positive view of Y's trustworthiness. If we take an agent's attitude toward another agent to be a belief about that agent, then: 'X trusts Y' is equal to the definition that 'X believes that Y is trustworthy'.

4. IoT TRUST MODELING

To demonstrate the proposed approach, let us consider a use case scenario where an entity A trusts an entity B (as *being reliable*) with a trust degree at least 0.8 and entity A trusts another entity C with a trust degree at least 0.2. Entities A, B and C are heterogeneous ('foreign' to each other) IoT entities that share however the same environment/context at a specific time interval, and all three are registered in a common publicly available IoT registry (high level IoT layer, at the information layer, e.g. an IoT ontology operating as a registry service).

Let us now explicate the scenario, placing the entities in the specific context of a smart room i.e. if motion is detected in the room then room's lamp is switched on. In this scenario, A is a smart application and B, C are motion detection sensors. A must be deployed in their common environment (the room) where B and C have already been deployed. There might be also the case where other entities of the same or different type (e.g. other smart devices), have also being deployed. In such a case, entity A cannot 'decide' which one of the entities matching the required specifications (based on the Smart Proxy computation of alignments of their specifications) is most appropriate to be used for the execution of its functionality: In this example, which motion detection sensor to select. For this reason, an automated

deployment of the application cannot be ensured (if decision cannot be made). However, by providing a degree of trustworthiness between IoT entities at the higher level of abstraction it is possible to *ensure* that the deployment of IoT entities in such scenarios will be performed: entity A will select the entity with the higher trust value among all matching entities in its context. Such value/degree of trust may be computed using a function that takes into account environmental/contextual information as well as other related information e.g. who the provider and owner of the entity is, what are the security policies of this entity, what are the previous deployment statistics of the entity, etc. In addition, such a trustworthy deployment, can be considered *secure,* since it involves the most trustworthy entities from the available (matched) ones within the deployment environment/context.

As an alternative scenario, we could think of a conference room context where a 3rd-party generic broadcasting application is 'searching' for the most trustworthy (most reliable in the specific context of the conference room) recording devices (microphones, cameras, smart phones with embedded capabilities) of registered visitors, before deploying itself in the environment.

For the demonstration of the proposed modeling approach, the IoT ontology and the automated deployment process of IoT entities presented in Kotis et al, 2012 [6] will be used. As already stated, in this work we extend IoT ontologies with *trust semantics*, and this is achieved by reusing only the main class of any IoT ontology i.e. 'IoT-entity' class.

4.1 The Smart Room scenario

In a smart room context, the following IoT entities have been registered (in the IoT-ontology):

- A smart room application (SmartRoomApp) which is capable of controlling lights in a room, based on the sensing of a motion detector,
- Two motion detection sensors provided by different vendors, A and B (using different namespaces for specifying their semantics) and owned by different agents (namely, 'Me' and 'Her'),
- Two smart lamps (a lamp attached to a smart switch) also provided by those two providers and owned by the same agents.

According to previous work [6], the task of matchmaking of entities' specifications, as part of the overall Smart Proxy solution in the Semantic Smart Gateway Framework (SSGF), should align and match the semantics of the registered entities, facilitating such way the communication of the application (via message translation) with the appropriate entities. However, in our scenario we've put more than one entity of the same type for an application to function, and we assume that all entities of the same type have the same matching score in the specifications' matchmaking output of the Smart Proxy. In the open IoT, where a large number of applications and devices will be registered in different publicly available IoT registries, such a scenario is more than likely to be seen in a quite larger scale (hundreds of devices of the same type and functionality can be possibly used by a generic third-party application within the same environment/context).

So, the question to answer in this scenario, which is the main concept of our work, is: which of the matched entities (motion detection devices in this case) the application must use to execute its logic? We conjecture that the key to this answer is 'trustworthiness' as in real life, where humans, based on who they trust more or less, choose to be coupled only with a subset of those who they possibly match with, or choose to buy only from a specific seller among those who provide exactly the same products and prices.

In the following paragraphs we present our solution based on the notion of *trust semantics* added in the IoT domain. Such semantics are provided as the key to an IoT application/service to select, among the available devices the most suitable ones, i.e. the ones that the application trust more than others. For demonstration reasons we use specific example namespaces at the following ontology IRIs:

- IRI of IoT ontology: http://purl.org/IoT/iot, prefix: iot
- IRI of IoT trust ontology: http://purl.org/IoT/iot-trust, prefix: iot-trust
- IRI of IoT application example domain ontology: http://purl.org/IoT/iot-app, prefix: iot-app
- IRI of IoT device provider A: http://purl.org/IoT/iot-provA, prefix: iot-provA
- IRI of IoT device provider B: http://purl.org/IoT/iot-provB, prefix: iot-provB

4.2 The IoT Trust Model

As already stated, in this work we extend IoT ontologies with *trust semantics*, and this is simply achieved by reusing the main (and most common) class of any IoT ontology: 'iot:IoT-Entity'. Our simple model introduces a binary relation between two IoT entities ('iot:IoT-entity') using the object property 'iot-trust:trusts' and its inverse ('owl:inverseOf') property 'iot-trust:trustedBy'. In addition, the model introduces such a property as a non-hierarchical fuzzy associative relationship, using fuzzyOwl2 semantics.

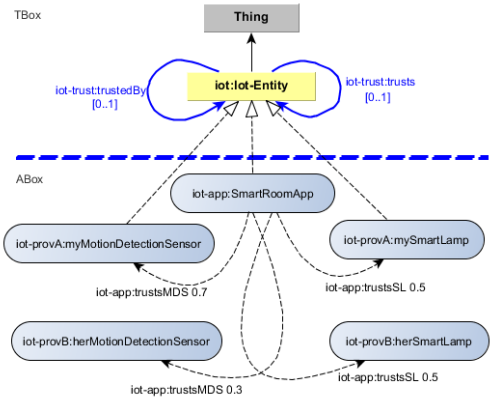

Fig. 3. The simple trust model for IoT entities

A graphical representation of the model is depicted in Figure 3, instantiated with IoT entities taken from our smart room scenario. Specific instantiations of the trust property i.e. trusts Motion Detection Sensor ('iot-app:trustsMDS') and trusts Smart Lamp ('iot-app:trustsSL') can be defined as sub-properties of iot-trust:trusts. The proposed simple model is depicted in the TBox area of Figure 3, where the instantiations of the model's entities using the example scenario are placed in the ABox. Specific degrees of trust between IoT entities are specified at the specific sub-properties of 'iot-trust:trust' property, i.e. at the 'iot-app:trustsSL' and 'iot-app:trustsMDS' respectively.

As stated, we've used fuzzy semantics representation of the 'iot-trust:trusts' object property in order to capture the degree of confidence (for an entity to be reliable) in the interval of [0,1]. Using the fuzzyOwl2 plugin of Protégé ontology engineering

environment, we are able to translate the instantiated example model in the well-known FuzzyDL [7] representation language:

(define-modifier trustModifier linear-modifier(1.0))
(define-primitive-concept IoT-entity *top*)
(inverse trustedBy trusts)
(domain trustedBy IoT-entity)
(domain trusts IoT-entity)
(range trusts IoT-entity)
(range trustedBy IoT-entity)
(related SmartRoomApp herSmartLamp trustsSL 0.5)
(related SmartRoomApp mySmartLamp trustsSL 0.5)
(related SmartRoomApp herMotionDetectionSensor trustsMDS 0.3)
(related SmartRoomApp myMotionDetectionSensor trustsMDS 0.7)
(implies-role trustsMDS trusts 1.0)
(implies-role trustsSL trusts 1.0)

In order to deliver however an extensible representation of our proposed simple model (Figure 3), and to fully comply with O'Hara's trust definition, we have further worked towards a more elaborated representation (Figure 4), by introducing the main class 'iot-trust:TrustworthinessObject' (TO) as the domain class of object properties 'iot-trust:has_trustor' and 'iot-trust:has_trustee'. Both properties have 'iot:IoT-Entity' as a range class. Furthermore, behavior is represented by the OWL class 'iot-trust:Behavior', being the range class of fuzzy object property 'iot-trust:has_behavior'. The TO is also related via the 'iot-trust:has_service' property with a specific service (represented in the IoT ontology) in order to capture the task(s) of the deployed application in a particular context e.g. a motion detection service with tasks/functionality to a) detect movement, b) switch on the lights. Finally, context is represented by the OWL class 'conon:ContextEntity', being the range class of 'iot-trust:has_context' object property. As a context-related namespace we have used (for demonstration reasons) the Context Upper ontology CONON [13] (prefix conon), but any other related class from a context domain-specific ontology can be used. Based on this definition of TOs, additional properties can be defined towards extending the model, such as the trust algorithm ('iot-trust:TrustAlgorithm') for computing trust values (via the 'iot-trust:trust_algorithm' object property). Any other possibly useful property related to the trustworthiness of an IoT entity pair may be easily added by specifying 'iot-trust:TrustworthinessObject' as its domain class.

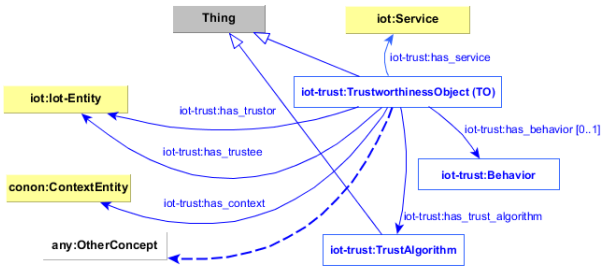

Fig. 4. The extensible IoT trust model

In this extended model, 'iot-trust:reliable' is defined as an instance of 'iot-trust:Behavior' (instance reliable Behavior 1.0), and TOs are instantiated with this behavior via fuzzy annotations of property 'has_behavior' (related TO_1 reliable has_behavior 0.5).

The extensible proposed model is engineered in a way that is capable of answering queries such as "For a *room* context, for a *smart room* application and for a *detection service* service/task, get the most *reliable entities* for its deployment". A trustworthiness/reliability threshold is set (e.g. 0.7) in order to filter the matched triples. An example query encoded in SPARQL (using a datatype property for the shake of simplicity in the presentation) is provided below:

*SELECT * WHERE {*
?trustObject a iot-trust:TrustworthinessObject.
?trustObject iot-trust:has_context conon:room.
?trustObject iot-trust:has_trustor iot-app:smartRoomApp.
?trustObject iot-trust:has_behavior iot-trust:reliable.
?trustObject iot-trust:has_service iot:motionDetectionService.
?trustObject iot-trust:hasTrustValue ?value.
FILTER (?value >=0.7)}

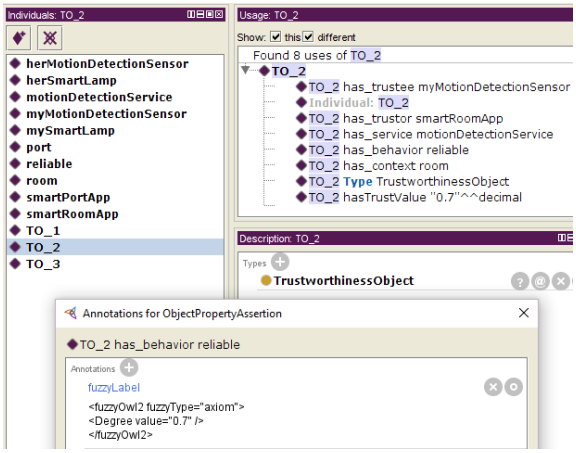

Fig. 5. TO instantiation example using 'reliable' as behavior type, 'room' as context and 'motionDetectionService' as service

The fuzzy query related to the behavior property of the example query are:

(min-related? TO_2 reliable has_behavior)

//Is TO_2 related to reliable through has_behavior ? >= 0.7

Further engineering of the ontology using fuzzy semantics can enrich the definitions in our model, for instance, by realizing entities as fuzzy members of specific classes: e.g. *which lamp is the most trustworthy instance of the class SmartLamp, given its specific characteristics*.

//definitions

(instance mySmartLamp SmartLamp 0.7)

(instance herSmartLamp SmartLamp 0.3)

//queries

(min-instance? herSmartLamp SmartLamp)

(min-instance? herSmartLamp SmartLamp)

//reasoner translation and answer

Is herSmartLamp instance of SmartLamp ? >= 0.3

Is mySmartLamp instance of SmartLamp ? >= 0.7

A working version of the extended IoT-trust ontology in OWL and FuzzyDL serialization, as well as the instantiated simple one introduced in section 4.2, can be accessed at http://ai-group.ds.unipi.gr/kotis/ontologies/IoT-trust-ontology.

5. DISCUSSION AND FUTURE WORK

Following O'Hara's definitions on context-depended trustworthiness, we accentuate the local (context-depended) range

of trust value computation as highly significant for IoT environments, focusing less on the general trustworthiness of IoT entities. A device (e.g. a video camera, a microphone, a phone with embedded camera and microphone) may be more reliable (thus more trustworthy) in a specific environment C (e.g. in a conference room) than in another (e.g. outside spot under sun, near sea and traffic) based on environmental conditions (e.g. sun, noise, moisture levels) that affect the function and consequently the reliability of the device and the performance of the application (unreliable and misbehaving devices minimize applications' performance).

In an extended line of this research, we are investigating an approach for dynamic trust management for a community-based social IoT environment by considering multiple social relationships among device owners [14]. In this work, a social IoT environment with no centralized trust authority is considered, introducing social relationships such as *ownership*, *friendship, community*. In such a work, we have designed the extension of the computation of the degree of trust by a context-depended property, called *capacity*. We define *capacity* as the ability of an IoT entity (a device or an application) to function within specific context requirements (e.g. environmental properties such as light, noise, temperature). Such requirements are specified in the IoT ontology (semantic registry) at the context level definition, and matched against devices' and applications' specs (also specified in the IoT ontology during their registration in the semantic registry). Such a matching task results to a capacity signature *cap* of an IoT entity *E* for a specific context *C*, i.e. to a capacity value for each device per context. Such a signature then is taken into consideration for the computation of trust value between two IoT entities. Issues such as the propagation (transivity) and aggregation of trust i.e. how to disseminate and combine trust information, are treated by a computational model, such as the one presented in [14]. Our extension of this particular computation model is under implementation using the NS-3-based[1] simulation system provided to us by its developers Bao & Chen 2012 [14].

In this paper we have presented a simple and extensible trust model that is seamlessly integrated in IoT ontologies, towards semantically enabling IoT trust for ensuring IoT entities' effective deployment in specific contexts. The work presented in this paper is focusing on IoT trust modeling, reusing existing trust models/ontologies as well as a framework for fuzzy semantics.

Future plans include a) the NS-3-based simulation and evaluation of a scalable method for computing context-based trust with no centralized trust authority, extending state-of-the-art well-defined and evaluated approach on dynamic trust management for community-based social IoT environment, b) a use case implementation and evaluation of the overall approach in real IoT setting (video conferencing broadcasting app and related sensors deployed on camera/mic-enabled mobile phones of socially-networked attendants in outdoor and indoor social meetings), taking into account information such as who the provider and owner of the entity is, what are the security policies of this entity, what are the previous deployment statistics of the entity, etc. Other issues concern the distribution of IoT-entities' information (context, app and devices properties, trustworthiness), in the absence of a central IoT registry or trustworthiness authority, utilizing social-networking infrastructure.

[1] https://www.nsnam.org/ns-3-13/

6. REFERENCES

[1] P. Barnaghi, W. Wang, C. Henson, and K. Taylor. Semantics for the Internet of Things: Early Progress and Back to the Future. *Int. J. Semant. Web Inf. Syst.* 8, 1 (January 2012), 1-21. DOI=http://dx.doi.org/10.4018/jswis.2012010101.

[2] S. Sicari, a. Rizzardi, L. a. Grieco, and a. Coen-Porisini. Security, privacy and trust in Internet of Things: The road ahead, Comput.Networks, vol. 76, pp. 146–164, Nov. 2014.

[3] R. Roman, J. Zhou, and J. Lopez. On the features and challenges of security and privacy in distributed internet of things, Comput. Networks, vol. 57, no. 10, pp. 2266–2279, Jul. 2013.

[4] Z. Yan and P. Zhang, A. Vasilakos. A Survey on Trust Management for Internet of Things, J. Netw. Comput. Appl., vol. 42, no. 2, pp. 120–134, 2014.

[5] Md. Mahmud Hossain, Maziar Fotouhi, and Ragib Hasan. Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things, IEEE Services Visionary Track on Internet of Things, New York, USA, June 2015.

[6] K. Kotis, and A. Katasonov. Semantic Interoperability on the Internet of Things: The Semantic Smart Gateway Framework, International Journal of Distributed Systems and Technologies (IJDST), vol. 4, issue 3, pp. 47-69, 07/2013.

[7] F. Bobillo, U. Straccia, Fuzzy ontology representation using OWL 2, International Journal of Approximate Reasoning, 52 (7), October 2011, Pages 1073–1094.

[8] Huang, J., and Fox, M.S., (2006). An Ontology of Trust - Formal Semantics and Transitivity, Proceedings of the International Conference on Electronic Commerce, Association of Computing Machinery, pp. 259-270.

[9] L. Viljanen. Towards an ontology of trust. In Trust, Privacy, and Security in Digital Business, pages 175–184. Springer, 2005.

[10] D. Ceolin, A. Nottamkandath, W.J. Fokkink and V. Maccatrozzo. Towards the definition of an ontology for trust in (Web) data, in Proc. 10th Workshop on Uncertainty Reasoning for the Semantic Web - URSW'14, Riva del Garda, CEUR Workshop Proceedings 1259, pp. 73-78.

[11] R. Alnemr, A. Paschke, and C. Meinel. Enabling reputation interoperability through semantic technologies. In I-SEMANTICS, pages 1–9. ACM, 2010.

[12] K. O'Hara. A General Definition of Trust. Technical report, University of Southampton, 2012.

[13] X. H. Wang, Zhang D., Gu T., Pung, H.K., Ontology based context modeling and reasoning using OWL, Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004, pp.18 – 22.

[14] F. Bao and I. Chen. Dynamic trust management for internet of things applications. In *Proceedings of the 2012 international workshop on Self-aware internet of things* (Self-IoT '12). ACM, New York, NY, USA, 1-6. DOI=http://dx.doi.org/10.1145/2378023.237.