Policy Management and Enforcement Using OWL and SWRL for the Internet of Things

Rustem Dautov[1,2(✉)], Symeon Veloudis[2], Iraklis Paraskakis[2], and Salvatore Distefano[1,3]

[1] Higher Institute of Information Technology and Information Systems (ITIS), Kazan Federal University (KFU), Kazan, Russia
{rdautov,s_distefano}@it.kfu.ru
[2] South-East European Research Centre (SEERC), The University of Sheffield, International Faculty CITY College, Thessaloniki, Greece
{rdautov,sveloudis,iparaskakis}@seerc.org
[3] University of Messina, Messina, Italy
sdistefano@unime.it

Abstract. As the number of connected devices is exponentially growing, the IoT community is investigating potential ways of overcoming the resulting heterogeneity to enable device compatibility, interoperability and integration. The Semantic Web technologies, frequently used to address these issues, have been employed to develop a number of ontological frameworks, aiming to provide a common vocabulary of terms for the IoT domain. Defined in Web Ontology Language – a language based on the Description Logics, and thus equipped with the 'off-the-shelf' support for formal reasoning – these ontologies, however, seem to neglect the built-in automated reasoning capabilities. Accordingly, this paper discusses the possibility of leveraging this idle potential for automated analysis in the context of defining and enforcing policies for the IoT. As a first step towards a proof of concept, the paper focuses on a simple use case and, using the existing IoT-Lite ontology, demonstrates different types of semantic classification to enable policy enforcement. As a result, it becomes possible to detect a critical situation, when a dangerous temperature threshold has been exceeded. With the proposed approach, IoT practitioners are offered an already existing, reliable and optimised policy enforcement mechanism. Moreover, they are also expected to benefit from support for policy governance, separation of concerns, a declarative approach to knowledge engineering, and an extensible architecture.

Keywords: Internet of Things · Semantic Web · Policy management · Policy enforcement · Web Ontology Language · Semantic Web Rule Language · Reasoning

1 Introduction

The unparalleled development of Information and Communication Technologies (ICT) in recent years has triggered the digital revolution, which is characterised by ubiquitous connectivity, leading to a convergence of technologies across

A. Puliafito et al. (Eds.): ADHOC-NOW 2017, LNCS 10517, pp. 342–355, 2017.
DOI: 10.1007/978-3-319-67910-5_28

different domains [8]. Rapid advances in embedded systems (especially mobile devices), wireless sensors, and mobile communications have led to the development of more and more powerful and capable personal devices. Modern tablets and even smartphones can be compared to 4–5 years old computers in terms of processing capabilities, so they can be also considered as effective computing systems. Indeed, they are more and more often taken into account as a laptop (and sometime desktop) replacement. This also contributed to their exponential growth in number and wide adoption; recent statistics [1] report on more than 2 billion smartphone users already connected to the Internet (i.e. every third smartphone owner), and quite soon, by 2020, this number is expected to reach 3 billion (i.e. every second smartphone owner). At the same time, similar statistics refers to the decreasing number of computers and indicates a reverse trend – i.e. there were almost 2 billion PC users in 2014, whereas nowadays their number has decreased to 1.5 billion, and is expected to drop to 1 billion by 2020.

Furthermore, new emerging ICT paradigms such as the Internet of Things (IoT), Cloud and Fog/Edge computing, enable novel, previously-unseen application scenarios for these devices. Smart Cities, Mobile Crowdsourcing, Digital Democracy, Citizen Science are just few simple examples among the wide range of possible scenarios, where a user can be directly and actively involved in everyday socio-technical activities by, for example, interacting through his/her personal device with the surrounding environment (e.g. smart lights, cameras, billboards, appliances, etc.) or with a remote entity (e.g. public administration).

These personal devices and their underlying infrastructure represent a 'digital image' of a user, acting as an interface between the user and the surrounding physical world, by measuring geo-localised physical data, such as weight, speed, acceleration, humidity, illumination, etc. (depending on the sensors available on-board). The massive amount of data captured in this context has laid the groundwork for the Internet of Services (IoS), which makes use of the context-aware data to provide Web services (typically in the form of Software-as-a-Service offerings) in order to generate value. The fusion between devices and data is characterised by complex networked collaboration between information sources, software, mechanical and electronic components interacting with physical entities. Today, such complex cyber-physical systems provide the technological foundation for diverse domain- and sector-specific applications, ranging from automotive and civil infrastructure, to healthcare, manufacturing, and transportation.

As a result, a heterogeneous ecosystem, constituted by embedded devices, sensors, actuators, mobile phones, and other smart connected objects, has been established through the Internet, as envisioned by the IoT. This technological fusion starts resembling a global 'melting pot', in which complex heterogeneous technologies provide solutions to 'little-local' problems. More specifically, existing solutions adopted a 'vertical' approach, in which ICT 'silos' are based on ad-hoc infrastructures, services and applications, narrow-tailored to specific problems at hand. This unscalable pattern relies on dedicated infrastructure, underpinned by specific hardware and software stacks, which will render unsustainable if, as expected, the demand for the IoT will keep increasing in the near future.

Taken together, these considerations identify the IoT as a rather fragmented 'archipelago' of isolated, local 'islands' of sensors and actuators, (static and mobile) devices (e.g. cameras, environmental sensors, personal portable devices, vehicles, etc.), network facilities (e.g. WiFi, 3 G/4 G, fiber optic, routers, base stations, cells, etc.), basic mechanisms and services for data management (i.e. collection, storage, aggregation, fusion, processing) – all related to and mainly conceived for a corresponding 'vertical' application domain. In this light, existing IoT systems seem to exist in isolation from each other – i.e. with little (or none) potential for integration, they are hardly compatible, interoperable or reusable.

1.1 Research Context

While challenges associated with timely processing of sensor data have been relatively successfully tackled by the advances in networking and hardware technologies [3,14], the challenge of properly handling data representation and semantics of IoT descriptions and sensor observations is still pressing. In the presence of multiple organisations for standardisation, as well as various hardware and software vendors, overcoming the resulting heterogeneity remains one of the major concerns for the IoT community. Moreover, apart from the syntactic heterogeneity (i.e. heterogeneity in the data representation, such as, for example, differences in data formats/encodings), heterogeneity in the semantics of the data can also be distinguished [6]. For example, different units of measurement, metric systems, or human languages are common causes for incompatibility and integrity problems in ICT.

As a potential solution to the IoT heterogeneity problem and yet-to-come standards, the research community started investigating potential ways of creating descriptive languages, which would provide an overarching modelling framework and bridge formerly-disjoint heterogeneous IoT systems at the semantic level. Such modelling languages can be seen as common vocabularies of terms, which are expected to be used by IoT practitioners to enable compatibility and interoperability when integrating IoT solutions. Simply put, two disjoint system would be able to 'communicate' to each other by expressing their data and interfaces, using a common suitable modelling language.

A representative example of how this lack of unified data representation has been addressed in the context of the IoT is the *Semantic Sensor Web* – a promising combination of two research areas, the Semantic Web and the Sensor Web [19]. Using the Semantic Web technology stack to represent data in a uniform and homogeneous manner, it provides enhanced meaning for sensor descriptions and observations so as to facilitate data analysis and situation awareness [7]. One of the main outcomes of this initiative was the Semantic Sensor Network (SSN) ontology – a thorough vocabulary, modelling the domain of physical sensor networks and observations, jointly developed by a wide group of researchers.

A number of other similar ontologies, modelling the IoT domain, have emerged in the recent years (for example, [2,4,9,11–13,15,16,18,23,25,26], to name a few). On the one hand, these ontologies are expected to provide common semantic foundation for describing various aspects of the IoT domain at

different granularity levels to address the IoT heterogeneity problem. On the other hand, however, the fact that there are more and more competing, disjoint ontologies serves as yet another evidence of the isolation of existing IoT systems from each other.

Moreover, from this perspective, Semantic Web ontologies do not differ much from other modelling approaches, such as, for example, Unified Modelling Language (UML) or Extensible Mark-up Language (XML) – i.e. they all may serve to provide taxonomies of terms and relationships, to be used as 'building blocks' when describing the IoT-related application domains. A major advantage of the Semantic Web languages, which is frequently neglected in this context, is the support for *automated formal analysis*, underpinned by the built-in reasoning capabilities of Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL), which are the key enabling technologies for the Semantic Web [10]. By representing data in terms of OWL classes and properties, one can perform reasoning tasks over these data and benefit from an already existing, highly-optimised and reliable analysis mechanism.

In this light, this paper is trying to tap into this idle potential for automated reasoning, and presents an approach to policy management and enforcement in the IoT context, using existing IoT ontologies and corresponding reasoning support. As it will be further described below, the proposed approach is expected to benefit from separation of concerns, extensibility, as well as increased opportunities for reuse, automation and reliability.

Accordingly, the rest of the paper is organised as follows. Section 2 contains relevant background information and briefs the reader on the foundations of the Semantic Web languages, as well as presents the target IoT-Lite ontology in more details. Section 3 presents and explains the proposed approach with a sample use case scenario, based on the IoT-Lite ontology. Section 4 summarises and discusses the main potential benefits of the approach. Section 5 concludes the paper.

2 Background: Semantic Web Languages

The Semantic Web, introduced by Berners-Lee [5] in 2001, is the extension of the World Wide Web that enables people to share content beyond the boundaries of applications and websites [10]. This is typically achieved through the inclusion of semantic content in web pages, which thereby converts the existing Web, dominated by unstructured and semi-structured documents, into a web of meaningful machine-readable information. Accordingly, the Semantic Web can be seen as a giant mesh of information linked up in such a way as to be easily readable by machines, on a global scale. It can be understood as an efficient way of representing data on the World Wide Web, or as a globally linked database. As shown in Fig. 1, the Semantic Web is realised through the combination of certain key technologies [10]. These technologies from the bottom of the stack up to the level of XML have been part of the Web standardised technology stack even before the emergence of the Semantic Web, whereas the upper, relatively new technologies – i.e. Turtle and N3, Resource Description Framework

(RDF), RDF Schema (RDFS), SPARQL Protocol and RDF Query Language (SPARQL), OWL, and SWRL – are intrinsic to the Semantic Web domain. All of these components have already been standardised by the World Wide Web Consortium (W3C) and are widely applied in the development of Semantic Web applications. The presented research specifically focuses on OWL and SWRL as the two potential ways of defining and enforcing policies in the context of the IoT, as it will be further explained in Sect. 3.

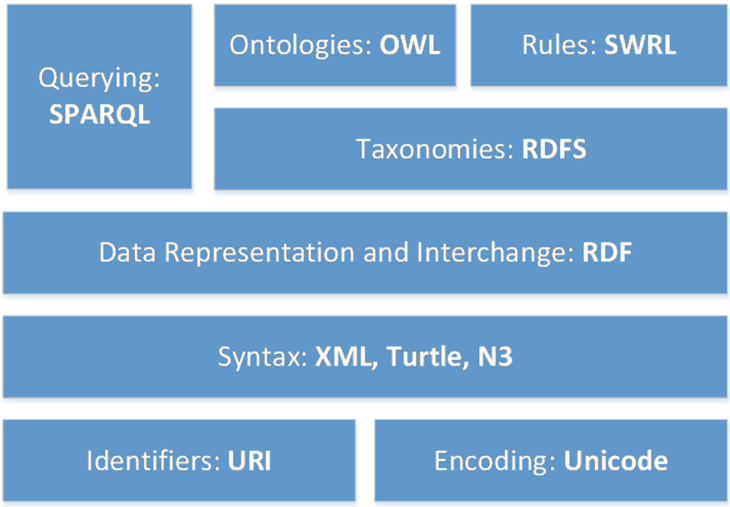

Fig. 1. The semantic web technology stack.

2.1 Web Ontology Language

OWL is a family of knowledge representation languages used to formally define an ontology – "a formal, explicit specification of a shared conceptualisation" [22]. Typically, an ontology is seen as a combination of a terminology component (i.e. TBox) and an assertion component (i.e. ABox), which are used to describe two different types of statements in ontologies. The TBox contains definitions of classes and properties, whereas the ABox contains definitions of instances of those classes. Together, the TBox and the ABox constitute the knowledge base of an ontology.

OWL provides advanced constructs to describe resources on the Semantic Web. By means of OWL it is possible to explicitly and formally define knowledge (i.e. concepts, relations, properties, instances, etc.) and basic rules in order to reason about this knowledge. OWL allows stating additional constraints, such as cardinality, restrictions of values, or characteristics of properties such as transitivity. The OWL languages are characterised by formal semantics – they are

based on *Description Logics* (DLs) and thus bring reasoning power to the Semantic Web. There exists a prominent visual editor for designing OWL ontologies, called Protege,[1] and several automated reasoners written in multiple programming languages, such as Pellet,[2] FaCT++,[3] and HermiT.[4] Depending on the expressive power, the OWL family of languages can be classified into *OWL-Lite* (the most light-weight, but least expressive), *OWL-DL* (more expressive, but still with automated reasoning support), and *OWL-Full* (most expressive, but undecidable, and therefore does not have reasoning support).

2.2 Semantic Web Rule Language

SWRL extends OWL with even more expressiveness, as it allows defining rules in the form of implication between an antecedent (i.e. body) and consequent (i.e. head). It means that whenever the conditions specified in the body of a rule hold, then the conditions specified in the head must also hold. It is worth noting, that fully compatible with OWL-DL, the SWRL syntax is quite expressive, which may have certain negative impacts on its decidability and computability. The sample code in Listing 1.1 contains a rule, expressed in a human-readable syntax, and illustrates the functionality of SWRL. The following sample states that if a city is located in England, then it is also located in the United Kingdom.[5]

Listing 1.1. Example of a SWRL rule.

```
City(?city) AND
    hasLocation(?city, http://en.wikipedia.org/wiki/England) THEN
hasLocation(?city, https://en.wikipedia.org/wiki/United_Kingdom)
```

Note that with OWL and SWRL, there is typically more than one way of defining knowledge to deduce same facts. For example, the above inference, drawn by reasoning over the SWRL rule, can also be achieved by defining `hasLocation` as a transitive property between `city` and `England`, as well as between `England` and the `United Kingdom`. Even though it is not explicitly stated that a city is located in the United Kingdom, the reasoner will deduce this fact, based on the knowledge that England is located in the United Kingdom by following the transitive property. In general, the SWRL reasoning is more computationally expensive than the OWL reasoning [20], making the latter a more preferable option in most cases.

[1] http://protege.stanford.edu/.
[2] https://github.com/complexible/pellet.
[3] https://code.google.com/p/factplusplus/.
[4] http://hermit-reasoner.com/.
[5] In this example, England and the United Kingdom are uniquely represented by their respective Wikipedia URLs.

2.3 An Existing Ontology: IoT-Lite Ontology

IoT-Lite ontology[6] [4] is a light-weight instantiation of the SSN ontology, actively developed by the W3C. It describes the key IoT concepts to allow interoperability and discovery of devices, sensors and sensor data in heterogeneous IoT platforms. This ontology reduces the complexity of other IoT models by describing only the main concepts of the IoT domain. This means that the IoT-Lite ontology can be extended by different models to increase its expressiveness and provide more focused modelling concepts if/when needed. That is, by following the Semantic Web principles of linking and reusing existing ontologies and datasets, it is possible to extend the core vocabulary with other relevant concepts, defined in other ontologies. This way, ontology engineers can simply import an existing, established, and trusted ontology, instead of 're-inventing the wheel' and developing their own, yet another, ontology from scratch.

3 Proposed Approach

Taking into consideration the presented features of the Semantic Web, this paper proposes leveraging reasoning capabilities of OWL and SWRL and utilise existing ontologies, describing the IoT domain, to enable creation and modification of policies to address a wide range of analysis activities in the IoT. This way, IoT practitioners can benefit from an already existing, optimised and reliable analysis engine, based on the declarative approach to defining the knowledge base.

The proposed vision will be now demonstrated thorough a sample use case scenario, based on the existing IoT-Lite ontology. Please note that the main goal of this sample scenario is to demonstrate how built-in reasoning capabilities can be used to perform analysis in the context of the IoT in a generic manner, rather than to focus on specific aspects of the IoT-Lite ontology. The proposed approach is expected to be universal and could be implemented using any other available IoT ontology. The sample pseudo-code snippets below are correspondingly simplified to make them easy to read and understand.[7]

The use case scenario focuses on a complex IoT system composed of multiple sensing devices, deployed both indoors and outdoors. Some of these devices are temperature sensors installed in rooms within a building. It is assumed that whenever any of these temperature sensors indicates a value exceeding a dangerous level of 50 degrees, the situation has to be classified as critical, and thus needs taking reactive actions. A possible way to handle this scenario would be

[6] https://www.w3.org/Submission/iot-lite/.

[7] The notations `ssn`, `iot`, and `dul` are established shortcuts for imported OWL ontologies, where corresponding concepts are defined.
SSN ontology (`ssn`): http://purl.oclc.org/NET/ssnx/ssn
IoT-Lite ontology (`iot`): http://purl.oclc.org/NET/UNIS/fiware/iot-lite
DOLCE Upper Level Ontology (`dul`): http://www.loa.istc.cnr.it/ontologies/DOLCE-Lite.owl.

to define explicit policies for every single temperature sensor within the building. In the worst case, such policies would be either 'hard-coded' with numerous `if/then` operators (i.e. any modifications would lead to the source code recompilation), or defined declaratively (i.e. stored in some kind of configuration file to be dynamically fetched by the analysis component). In both cases, however, the resulting knowledge base would be saturated by the excessive number of hardly manageable and possibly conflicting policies.

An alternative solution is based on using the built-in reasoning capabilities of OWL and SWRL to classify observed IoT data as instances of specific classes. More specifically, this use case demonstrates three different types of automated classification:

1. Defined Classes in OWL. Underpinned by the DLs, OWL allows creating so-called *defined classes* via a set of necessary and sufficient conditions. This means that the reasoner will classify any entity with a required set of sufficient properties as an instance of a specific class, even if this class membership was not defined explicitly. The defined class `RoomDevice` is demonstrated in Listing 1.2, which should be read as "if `sd` is a sensing device and has a rectangular coverage area, then `sd` is a device installed in a room".

Listing 1.2. Defined OWL class `RoomDevice`.

```
ssn:SensingDevice(?sd) AND
    iot:Rectangle(?r) AND
    iot:hasCoverage(?sd,?r) ≡
iot:RoomDevice(?sd)
```

2. Subclass Relationships in OWL. OWL also provides a simpler and more explicit way of defining classes and subclass relationships. It supports multiple and transitive inheritance, and, as in many other programming languages, subclasses inherit all the properties of their parent classes. The code snippet in Listing 1.3 contains two definitions. The first definition simply states that any temperature sensor is a sensing device. The second definition illustrates the transitive inheritance through a subclass hierarchy that states – in simple words – that if a device is installed in a room, then it is automatically assumed to be installed in a building as well, which in turn means it is an indoor device.

Listing 1.3. Defining OWL subclass relationships.

```
iot:TemperatureSensor IS A ssn:SensingDevice
iot:RoomDevice IS A iot:BuildingDevice IS A iot:IndoorDevice
```

3. Class Definition in SWRL. SWRL allows defining more expressive rules and takes the form of Horn-like rules, as illustrated by Listing 1.4. In simple words, the code snippet reads that if there is an indoor device `id`, indicating

that its measured value has exceeded 50 degrees, the current observation has to be classified as critical.

Listing 1.4. Defining the class `CriticalObservation` using the SWRL.

```
iot:IndoorDevice(?id) AND
    ssn:Observation(?o) AND
    dul:Value(?v) AND
    iot:observes(?id,?o) AND
    ssn:hasValue(?o, ?v) AND
    swrl:greaterThan(?v, 50) THEN
iot:CriticalObservation(?o)
```

Next, the presented use case scenario assumes that there is a temperature sensor `ts` reporting a temperature level of 60 degrees in its covered rectangular area. Taking together all three definitions above, the automated reasoner will take the following steps when resolving this situation[8]:

1. Since `TemperatureSensor` is a subclass of `SensingDevice`, `ts` is classified as `SensingDevice` (according to Listing 1.3).
2. Since `ts` is a `SensingDevice` and has a rectangular coverage area, it is classified as `RoomDevice` (according to Listing 1.2).
3. Since `RoomDevice` is a subclass of `BuildingDevice`, which is a subclass if `IndoorDevice`, `ts` is classified as an instance of `IndoorDevice` (according to Listing 1.3).
4. Finally, since `ts` is an `IndoorDevice` and its measured observation is greater than 50 degrees, this observation is classified as `CriticalObservation` (according to Listing 1.4).

This way, the reasoning engine is able to identify a situation, when a dangerous level of 50 degrees has been exceeded – i.e. to detect a critical situation by inferring implicit information from the limited, explicitly provided facts. Moreover, it is worth explaining that using generic rules for a wide range of devices, as in the example above, does not affect the flexibility of the proposed approach and its ability to define fine-grained, targeted policies for individual devices. Apart from inheritance, the OWL also supports overwriting parent properties by subclasses. This means that it is possible to enforce device-specific policies, which will overwrite the default behaviour and apply only to those specific devices. This way, flexible policy enforcement at various granularity levels can be achieved.

4 Discussing the Potential Benefits

When defining monitoring and analysis policies with OWL and SWRL, IoT developers are expected to benefit from the following [6]:

[8] Please note that there are two main reasoning methods, which define the order, in which axioms are considered for evaluation – namely, *forward* and *backward* chaining [21]. In the presented use case, forward chaining is assumed to be in place.

Policy Governance. Ideally, a policy enforcement mechanism is expected to enable stakeholders to perform changes to policies and policy sets 'on the fly', i.e. in a dynamic manner that does not require re-compiling, re-deploying and restarting the entire software system. Such changes may include, for example, the introduction of new policies, or the update and retirement of existing ones. In contrast to other current approaches that tend to rely on hard-coded analysis logic, the semantic approach sufficiently addresses this requirement. In particular, by promoting a semantic representation of policies, that ontologically captures the various knowledge artefacts that are encoded in the policies, the semantic approach promotes a *separation of concerns* that disentangles the expression of policies from the actual code of the applications that enforce them. This not only enables the performance of 'on the fly' changes to policy sets, but also crucially paves the way for the construction of a novel *policy governance framework* that is capable of determining the consequences of such changes on the overall *effectiveness* of the policies through the provision of the following seminal capabilities:

- By enabling automated reasoning about the *correctness* of a newly created or updated policy by harnessing the various knowledge artefacts that it embodies;
- By enabling automated reasoning about potential *inter-policy relations*.
- By enabling the performance of *rule-based policy lifecycle management*.

With respect to the first capability, an *iterative process* that aims at defining suitable *ontological templates* for the expression of policies is advocated [24]. This process consists of a number of iterative refinement steps, where each step introduces new concepts and properties that reify the high-level concepts, and the properties thereof, that appear in the IoT-Lite ontology; the new concepts typically take the form of sub-concepts of the existing higher-level ones. The ontological templates that are ultimately defined through this process articulate all those concepts that *must*, *may* or *must not* be embodied in a policy, as well as the allowable values that these concepts may attain in a particular context of use. For instance, the iterative refinement process may reify the `geo:Point` concept of the IoT-Lite ontology with such concepts as `Building`, `Floor` and `Room`. An ontological template may then be defined to insist that a policy must invariably incorporate readings from a temperature sensor that is located in a particular room of a specific building or, alternatively, from one of the sensors that are located in a specific floor of the same building. Any newly created or updated policy that is derived as an instantiation of this template is guaranteed to satisfy this requirement and incorporate these readings. Evidently, the ontological templates enable stakeholders to influence the allowable form, or structure, that complex policies in the IoT domain may attain in a particular context of use. In this respect, they enable stakeholders to infuse into the policies their particular business logic.

With respect to the second capability, inter-policy relations such as subsumption and contradiction that are potentially brought about by changes in the policies (e.g. introduction of new policies or updates of existing ones) and which may affect the effectiveness of the policies, are determined through the use

of 'off-the-shelf' DLs reasoners. Finally, with respect to the third capability, a framework may be constructed that generically determines the conditions under which *policy lifecycle actions*, such as policy updates and retirements, may be performed.

Human Readability and Ease of Use. The Semantic Web research targets at making information on the Web to be both human- and machine-readable, with languages that are characterised by an easy-to-understand syntax, as well as the visual editors for effortless and straight-forward knowledge engineering. OWL ontologies are known to be used in a wide range of scientific domains (for example, see [17] for an overview of biomedical ontologies), which are not necessarily closely connected to Computer Science, and allows even for non-professional programmers (i.e. domain specialists) to be involved in the process of policy engineering.

Extensibility. IoT systems may be composed of an extreme number of smart devices spread over a large area (e.g. traffic sensors distributed across a city-wide road network) and have the capacity to be easily extended (as modern cities continue to grow in size, more and more sensors are being deployed to support their associated traffic surveillance requirements). To keep up with this rapid growth and address the scalability issue, the proposed semantic approach, using the declarative definition, can extend the knowledge base to integrate newly-added devices in a seamless, transparent, and non-blocking manner. The same applies to the reverse process – once old services are retired and do not need to be considered anymore, the corresponding policies can be seamlessly removed from the knowledge base, so as not to overload the reasoning processes.

Increase in Reuse, Automation and Reliability. Policy enforcement mechanisms already exist in the form of automated reasoners for the OWL/SWRL languages, and the proposed approach aims to build on these capabilities. Since the reasoning process is automated and performed by an existing reasoning engine, it is expected to be free from so-called 'human factors' and more reliable, assuming, of course, the validity of ontologies and policies. Arguably, as the policy base grows in size and complexity, its accurate and prompt maintenance becomes a pressing concern so as to avoid potential policy conflicts. It is also important to keep in mind that formal reasoning based on DLs is a relatively expensive computational task, typically requiring an increased amount of computational resources (especially in the presence of numerous IoT devices). In this light, the proposed approach assumes that the policy enforcement is supposed to take place on a central (cloud-based) server, responsible for monitoring the IoT observation streams and take reactive actions, if required.

5 Conclusion

This paper discusses the possibility of utilising the idle potential for automated formal reasoning of existing IoT ontologies in the context of policy enforcement. While multiple IoT ontologies have been proposed both by the industry and the academia, there seems to be no evidence of a policy enforcement mechanism developed on top of these existing ontologies. As it was explained, the Semantic Web languages are underpinned by the Description Logics, which offer reasoning support, and enables automated classification of IoT observations. This means that IoT engineers can use existing ontological classes and properties to define policies, and, as a result, benefit from the built-in reasoning-based policy enforcement mechanisms.

As demonstrated by the sample use case scenario, with the proposed approach it is possible to define a set of policies, using various expressive OWL and SWRL constructs. The resulting policies can be generic (i.e. apply to a wide range of IoT devices) or more fine-grained (i.e. apply to specific individual devices). Moreover, IoT practitioners are expected to benefit from separation of concerns and support for policy governance, extensibility, and increased opportunities for reuse, automation and reliability.

Acknowledgements. The work presented in this paper was partially supported by the ERASMUS+ Key Action 2 (Strategic Partnership) project IOT-OPEN.EU (Innovative Open Education on IoT: improving higher education for European digital global competitiveness), reference no. 2016-1-PL01-KA203-026471. The European Commission support for the production of this publication does not constitute endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

References

1. Number of smartphone users worldwide from 2014 to 2020 (2017). https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/. Accessed 14 July 2017
2. Agarwal, R., Fernandez, D.G., Elsaleh, T., Gyrard, A., Lanza, J., Sanchez, L., Georgantas, N., Issarny, V.: Unified IoT ontology to enable interoperability and federation of testbeds. In: Proceedings of 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 70–75. IEEE (2016)
3. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Comput. Netw. **38**(4), 393–422 (2002)
4. Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., Taylor, K.: IoT-Lite: a lightweight semantic model for the Internet of Things. In: Proceedings of 2016 International IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, pp. 90–97. IEEE (2016)
5. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The semantic web. Sci. Am. **284**(5), 28–37 (2001)

6. Dautov, R., Kourtesis, D., Paraskakis, I., Stannett, M.: Addressing self-management in cloud platforms: a semantic sensor web approach. In: Proceedings of the 2013 International Workshop on Hot topics in Cloud Services, pp. 11–18. ACM (2013)
7. Dautov, R., Paraskakis, I., Stannett, M.: Towards a framework for monitoring cloud application platforms as sensor networks. Cluster Comput. **17**(4), 1203–1213 (2014)
8. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gener. Comput. Syst. **29**(7), 1645–1660 (2013)
9. Gyrard, A., Serrano, M., Atemezing, G.A.: Semantic web methodologies, best practices and ontology engineering applied to Internet of Things. In: Proceedings of 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), pp. 412–417. IEEE (2015)
10. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC, Boca Raton (2009)
11. Hu, S., Wang, H., She, C., Wang, J.: AgOnt: ontology for agriculture Internet of Things. In: Li, D., Liu, Y., Chen, Y. (eds.) CCTA 2010. IAICT, vol. 344, pp. 131–137. Springer, Heidelberg (2011). doi:10.1007/978-3-642-18333-1_18
12. Kinkar, S., Hennessy, M., Ray, S.: An ontology and integration framework for smart communities. J. Comput. Inf. Sci. Eng. **16**(1), 011003 (2016)
13. Kotis, K., Katasonov, A.: Semantic interoperability on the web of things: the semantic smart gateway framework. In: Proceedings of 2012 Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), pp. 630–635. IEEE (2012)
14. Liang, S., Croitoru, A., Tao, V.: A distributed geospatial infrastructure for sensor web. Comput. Geosci. **31**(2), 221–231 (2005)
15. Nambi, A.U., Sarkar, C., Prasad, V., Rahim, A.: A unified semantic knowledge base for IoT. In: Proceedings of 2014 IEEE World Forum on Internet of Things (WF-IoT), pp. 575–580. IEEE (2014)
16. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the Internet of Things: a survey. IEEE Commun. Surv. Tutorials **16**(1), 414–454 (2014)
17. Rubin, D.L., Shah, N.H., Noy, N.F.: Biomedical ontologies: a functional perspective. Briefings in Bioinform. **9**(1), 75–90 (2008)
18. Seydoux, N., Drira, K., Hernandez, N., Monteil, T.: IoT-O, a core-domain IoT ontology to represent connected devices networks. In: Blomqvist, E., Ciancarini, P., Poggi, F., Vitali, F. (eds.) EKAW 2016. LNCS (LNAI), vol. 10024, pp. 561–576. Springer, Cham (2016). doi:10.1007/978-3-319-49004-5_36
19. Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. IEEE Internet Comput. **12**(4), 78–83 (2008)
20. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: a practical OWL-DL reasoner. Web Semant. Sci. Serv. Agents World Wide Web **5**(2), 51–53 (2007)
21. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations, vol. 13. MIT Press, Cambridge (2000)
22. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. Data Knowl. Eng. **25**(1–2), 161–197 (1998)
23. Toma, I., Simperl, E., Hench, G.: A joint roadmap for semantic technologies and the Internet of Things. In: Proceedings of the Third STI Roadmapping Workshop, vol. 1 (2009)

24. Veloudis, S., Paraskakis, I.: Defining an ontological framework for modelling policies in cloud environments. In: Proceedings of 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp. 277–284. IEEE (2016)
25. Wang, W., De, S., Toenjes, R., Reetz, E., Moessner, K.: A comprehensive ontology for knowledge representation in the Internet of Things. In: Proceedings of 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1793–1798. IEEE (2012)
26. Zhao, S., Zhang, Y., Chen, J.: An ontology-based IoT resource model for resources evolution and reverse evolution. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) ICSOC 2012. LNCS, vol. 7636, pp. 779–789. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34321-6_62