Journal of King Saud University – Computer and Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Towards a unified ontology for IoT fabric with SDDC

Koundinya Koorapati [a,b,*], Rubini Pandu [a], Prem Kumar Ramesh [c,d], Sairam Veeraswamy [e], Usha Narasappa [f]

[a] CMR University, Bangalore, India
[b] HCL Technologies, Bangalore, India
[c] Department of Computer Science and Engineering, CMR Institute of Technology, Bengaluru, India
[d] Visvesvaraya Technological University, Belagavi, Karnataka, India
[e] VMware, Bangalore, India
[f] HCL Technologies, Bangalore, India

ARTICLE INFO

Article history:
Received 30 October 2020
Revised 6 April 2021
Accepted 27 April 2021
Available online xxxx

Keywords:
Internet of Things (IoT)
Cloud computing
Software-Defined Data Center (SDDC)
Ontology
Semantic web

ABSTRACT

The ever growing Internet of Things (IoT) paradigm aims to put people's incorporation into the new phase of connectivity and sensor technology. While, IoT has the potential to deliver new value-added services in order to make life easier and healthier for people, there are still several issues to be addressed in order to harness the widespread dissemination and adoption of the IoT paradigm, considering its potential benefits. In this context, considering an IoT ecosystem holistically i.e. end-to-end which includes the proprietary Operational Technology (OT) comprising of software and hardware to monitor, detect and control the equipment through sensors and actuators and the general Information Technology (IT) which comprises of the data center infrastructure catering to the backend needs of IoT such as compute, storage and network, one major challenge is converging OT with IT. Such an OT/IT convergence has the potential to explore many problems that exist in the end-to-end IoT ecosystem. One such problem is related to drawing actionable insights through operational analytics. The data center paradigm which concerns this research is the Software-Defined Data Center (SDDC) which is touted as the preferred data center paradigm for IoT. While most of the research happening on IoT today deals with the data from the IoT applications and sensors, this research focuses on the plethora of the metadata contained in the end-to-end IoT ecosystem. This research demonstrates the modelling of the end-to-end IoT ecosystem using semantic-based approach such as ontology. We arrive at a cohesive unified ontology unifying the OT and the IT constituents of the IoT fabric with SDDC together with rich context aware attributes embedded in the unified ontology, thereby addressing the problem of OT/IT convergence as well as laying the foundation to explore many solutions to problems pertinent in such a converged OT/IT ecosystem.

1. Introduction

With the number of connected devices on the rise and IoT gaining rapid momentum, end-to-end IoT ecosystem having the modern day data center such as SDDC (**S**oftware-**D**efined **D**ata **C**enter) catering to the backend needs of IoT present many challenges. One such challenge is the increased need to converge the **O**perational **T**echnology (OT) consisting of the sensors and devices and **I**nformation **T**echnology (IT) consisting of the data center infrastructure (Christy, 2017). When we talk of SDDC, it is regarded as the cloud building block, and Cloud can also be an SDDC extension. For example, if we look inside the public cloud offering, we can find that the idea of the SDDC is quite similar. However, we could also have the "Software-Defined" at our own on-site data center with extensions to a public cloud. In SDDC, all functionality is automated and hardware-layer abstracted. The compute/CPU, network and storage can then communicate freely without the need for any particular hardware type. All interfaces between each software component are regulated by API.

Today, there exists no standard way to implement OT/IT convergence and thereby exploit the plethora of opportunities to solve

* Corresponding author.
E-mail addresses: Koundinya.15phd@cmr.edu.in, koundinya@hotmail.cm (K. Koorapati), Rubini.p@cmr.edu.in (R. Pandu), Premkumar.r@cmrit.ac.in (P.K. Ramesh), sveeraswamy@vmware.com (S. Veeraswamy), Usha.bn@gmail.com (U. Narasappa).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.04.015
1319-1578/

interesting problems for end-to-end IoT ecosystem. As a consequence, there arises a need to model the end-to-end IoT ecosystem with SDDC. Thus, this research is an attempt to find a solution for the convergence of OT and IT, and using the same to find solutions to problems such as managing the resources in the end-to-end IoT fabric, power profiling of the end-to-end IoT ecosystem, and carrying out predictive failure analytics hold the key. Although SDDC builds on virtualization of storage, compute and network with rich set of API for automation, which enables statistical multiplexing of resources across customers and IoT applications using the SDDC, questions such as: *How can we seamlessly bring together the OT and IT worlds?*, *What is the best data center configuration for a given IoT application? Can we make any recommendation based on the power drain of a sensor? How can we proactively predict failures in the IoT fabric and apply remediation steps?* lead into interesting problems for our research prevalent in the end-to-end IoT ecosystem. But before that, it is important to model the end-to-end IoT ecosystem with SDDC. To that end, the following are our key objectives in this research:

- Holistic perspective to the IoT stack by taking into consideration not just sensors, but also the data center elements such as compute, network and storage.
- Modeling of the end-to-end IoT ecosystem with SDDC using semantic based approach such as ontology.
- Convergence of the OT and the IT constituents of the IoT ecosystem and laying the foundations to solve interesting problems in the end-to-end IoT ecosystem.

Our paper is organized as follows. In Section 2, we shall familiarize the readers with some key concepts which are necessary for this research work. Section 3 explores related literature concerned with our work. The system architecture for our proposal is presented in Section 4. We look in detail the construction of our cohesive/unified ontology and its validation in Section 5. We highlight the importance of OT/IT convergence with the unified ontology in Section 6 by demonstration two examples. Finally, in Section 7, we recapitulate our contributions and brief some of the future work.

2. Background

In this section we shall briefly acquaint the readers with some of the concepts which are related to this research such as ontology and the importance of OT/IT convergence.

2.1. Ontology

In short, ontology is the systematic definition of knowledge or expertise as a collection of concepts in a field and relations between them (Gruber, 1995). In order to allow this sort of definition, in addition to constraints, regulations and axioms, we want to formally define individual components (instances), classes, attributes and relations. In addition, ontologies can add new details about the field and not just provide a shareable and reusable presentation of knowledge.[1] In the context of this research, ontologies are used to model the domain of IoT ecosystem with SDDC.

2.2. The need for OT and IT convergence

Some of the most compelling reasons for IT and OT convergence are (Chapple, 2015; Morliere, 2020):

2.2.1. Process convergence

OT has a set of personnel such as OT admins, technicians, managers, etc., whose job responsibilities include taking care of the OT side of things such as monitoring the sensors, keeping an inventory of the sensors so that the failed sensors can be replaced as quickly as possible, applying security policies to the sensors. In addition, IT has a range of staff, including IT managers, technicians, laboratory operators and IT managers whose duties include IT support for elements such as data center storage, compute, networking, and security in order to ensure that data centers are free of downtime. So, in a way, the roles and responsibilities of the personnel on the OT and IT side are more or less similar. Thus, convergence can bring more visibility into the IT and OT operations by having a common set of personnel who would be responsible for the end-to-end IoT ecosystem. More relevant Key Performance Indicators (KPIs) with common objectives can be derived which are more beneficial than both of them operating in individual silos.

2.2.2. Proximity convergence

OT assets and infrastructure such as sensors, actuators are usually widely distributed geographically closer to the source where the data has to be sensed or acted upon. Thus the OT personnel responsible for the OT operations need to be at the location to carry out operations such as support and upgrades. Although the staff remotely access the infrastructure with software tools to carry out such operations to reduce their need to travel to the site, the OT staff must also have an insight into the IT infrastructure so as to understand the impact of IT change. Similarly, we need to monitor the impact of OT change on IT. This means that OT-IT integration aims to improve monitoring of distributed operations. This in turn leads to improved structures and greater efficiency of the company. This also implies that productive workers work with better knowledge. Based on timely and reliable knowledge, operations managers benefit from better strategic decisions.

2.2.3. Security

Cyber security challenges have always been a challenge for the IT environments where the IT personnel are constantly working towards protecting the data center infrastructure from malware attacks. Likewise, for the OT personnel the increasing pervasiveness of connectivity has amplified the possibility [2] of OT networks having the sensors and devices to be attacked as well. According to the report from Forrester (Merritt, 2018), most of the organizations now have IoT Technologies which in some form or the other feed their data to the data centers run by the organization. IoT devices usually made from consumer-grade technologies, are unmanaged, plethora for vendors using non-standard software and operating systems, diverse range of insecure protocols, potential to connect to devices outside the organizations network, weak security practices like simple credentials, not adhering to encrypting traffic pose grave security issues. The devices individually can be attacked or the networks themselves be attacked through buffer overflows or denial of service attacks. Thus, IT and OT can benefit from each others learning of security concerns and come up a common set of security policies that can be applied, managed and maintained for the entire IoT ecosystem.

2.2.4. Complicated compliance requirements

Having the personnel visit the sites for auditing and monitoring the compliance for assets based on certain industry standards that have to be adhered by the OT and IT departments is at times very laborious due to the manual processes involved and can give room for errors. This also causes inefficiency due to more focus on the

[1] https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/.

[2] http://www.sans.edu/research/security-laboratory/article/did-attack-surface.

process compliance than useful work. Thus, automating the inventory and management of the assents on the IT and OT side using a common asset visibility model can help the organization cut labors costs and increase productivity. As an analogy, we come across compliance requirements which are complicated and analyzed in these works relevant to the legal reasoning domain (Olivieri et al., 2013; Olivieri et al., 2015; Scannapieco et al., 2013).

2.2.5. Data and software convergence

Instead of having individual applications on the IT and OT side, thanks to convergence, it would be possible to have applications that can work on the data from OT and IT thereby achieving software and data converge. Thus, it would be possible to have smart infrastructure analytics applications that carry out operational analytics in the converged ecosystem.

3. Related works

As more and more IT infrastructures of enterprises start adopting cloud computing with the objective of meeting the resource needs for storage, compute and network on cloud for the advantages which cloud has to offer such as scalability, flexibility, high availability, agility of resources, there still remains challenges for IT departments to fully realize the vision of an automated enterprise cloud. To accomplish this, the information of the enterprise cloud has to be managed efficiently. This is where the authors, Haase et al. (2010) demonstrate the use of semantic technologies like ontologies for managing information in the enterprise cloud infrastructure. Here, the authors citing the examples of Amazon AWS, Salesforce.com mention that the objective of a fully automatable data center can be realized. What is noteworthy is the mentioning of API driven tasks replacing the manual tasks in a data center which is in fact the modern data center or SDDC today. This is one of the earliest work during that time which foresees the automated data center in the form of an enterprise cloud. Their main contribution focuses on managing of infrastructure automatically and the challenges pertinent to this. This is where the authors demonstrate the capabilities of semantic technologies like ontologies in overcoming this challenge. The authors gives due consideration to*data integration, documentation and annotation, and intelligent information access and analytics*.

From Delicato et al. (2017), Flavia et al. (2017), one of the very important activity is to model an IoT ecosystem in a manner so that all the layers of the ecosystem are represented in such a way that managing of the resources in the ecosystem for any activities such as monitoring, estimating, allocating of the resources becomes easy and seamlessly favorable for developing innovative solutions to address the various challenges presented by IoT today. Resource modelling not only deals with the representation of the resources, but instead also concerns itself with the representation of application. Thus, as a part of literature review, we study three main categories of resource representation which are based on attributes, virtualization and semantics. We review in detail (Delicato et al., 2017), where the authors Flavia et al. (2017), present examples of various approaches for modelling resources for IoT and also draw a comparison indicating that semantic based approaches stand out when compared with other approaches.

The Table 1, gives an overview of the resource modelling approaches studied so far based on the various degree of characteristics for each approach. The degree of abstraction, granularity, formalism, expressiveness and flexibility are considered in our review as described in Delicato et al. (2017).

The paper (Youseff et al., 2008; Youseff et al., 2008) is one of the key works undertaken to resolve the very problem of interoperability with the Cloud. In constructing a coherent ontology for the general cloud computing phenomenon, the authors have offered a great reference. The purpose of the research was to provide a coherent picture of the ecosystem of cloud computing. Their work provides a systematic breakdown of the Cloud with five separate levels of cloud-based ontology. The five layers suggested in the research include the Cloud Application Layer, the Cloud software environment layer, the Cloud software infrastructure layer, the software kernel and the hardware and firmware, and the computing resources, storage and communication for the cloud infrastructure layer.

In the field of electronics, computer science and engineering, industrial engineering, automation has played a very important role to find solutions to everyday challenges. As automation is synonymous with SDDC, we review (Cristani et al., 2018) which proposes an ontology based solution for transforming PLC (programmable logic control) based automation plant to one that is driven by single board computers (IoT). This work demonstrates that with an ontology of the actions of the plant alone, the core approach of the methodology being able to abstract the details of implementation.

4. System architecture

As shown in Fig. 1, the architecture of our holistic end-to-end system consists of the OT (Operational Technology) infrastructure such as the sensors, actuators, etc., along with their associated communication protocols such as Wifi, ZigBee etc., and the IT (Information Technology) infrastructure comprising of the data center infrastructure. The IT infrastructure can either be on-premises (SDDC/private cloud) or off-premises (cloud).

The topmost layer is the cloud layer which can either be a public, private or hybrid cloud. The layer below cloud is the SDDC layer which is Core and which provides compute, network, storage and security either on-premises or off-premise (cloud). The layer below Core is referred to as Edge which comprises of the gateways that support various communication protocols indicated by the Communications Layer such as WiFi, Zigbee, LTE and so on thereby enabling the devices represented by the Devices layer such as sensors and actuators connect to the gateway.

5. Construction of cohesive ontology for IoT with SDDC

We choose to adopt a hybrid approach model for the integration of our ontologies which is in fact a combination of single and multiple approaches. The reasoning behind this being, for our ecosystem of IoT fabric with SDDC/Cloud as shown in the Fig. 2, we have the IoT ontology which has its own vocabulary and then we have some shared vocabulary for Cloud and SDDC ontologies. This shared vocabulary contains some of the terms (the primitives) of SDDC and Cloud which are used to build complex terms and relationships of the destination ontology. Constructing an ontology using the hybrid approach allows the flexibility to add new sources without requiring the modifications in the shared vocabulary of the mappings. This approach facilitates the acquisition and evolution of ontologies. Because the ontologies of the source refer to the shareable content/vocabulary, it can be a challenge to recycle the ontologies for our own use.

Having chosen the hybrid approach to integrate/unify our individual ontologies, we follow the iterative approach as given in Fig. 3 to construct the unified ontology depicted in layers show in Fig. 4.

We use the following iterative steps as proposed in Ismail et al. (2006):

1. **Design:** This process is used to establish the context and aim of the ontology. Additionally, the relationship among the classes and sub-classes are also scoped.

Table 1
Modelling approaches for IoT ecosystems.

Modelling approach	Degree of abstraction	Granularity	Formalism	Expressivity	Flexibility
Attribute based	Low	Fine	High	Low	Low
Virtualization based	High	Fine + Coarse	Low	Low	High
Semantic-based	High	Fine + Coarse	High	High	High

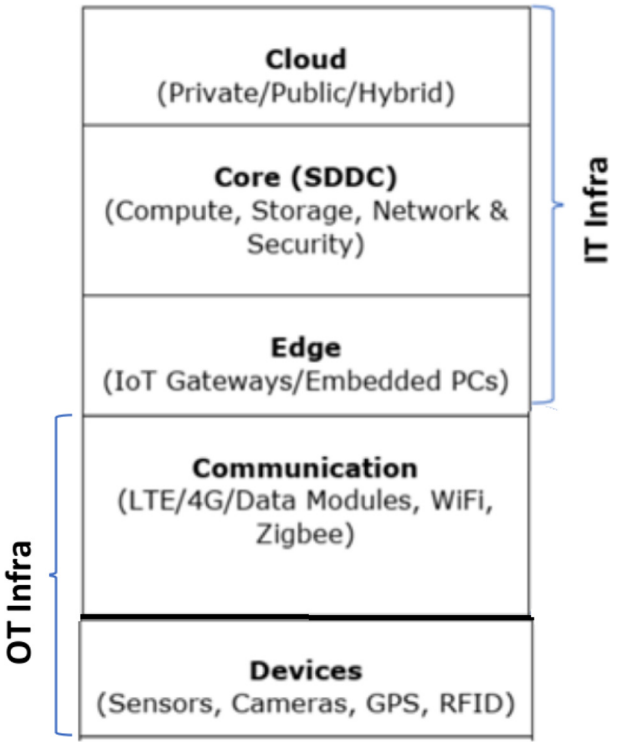

Fig. 1. IoT Fabric with SDDC showing OT and IT infrastructure components.

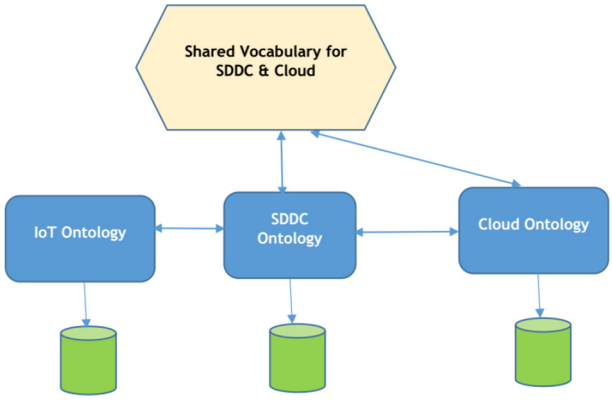

Fig. 2. Hybrid integration approach for ontologies.

2. **Implement/Develop:** This step is used to take a decision whether the ontology has to be implemented from scratch or reuse an existing ontology.
3. **Integrate:** This step unifies the ontology developed in **step 2** with an already existing ontology.
4. **Validation and Feedback:** This step is primarily used for seeking the opinion of the subject matter experts to validate the correctness of the completed ontology. The step also makes use of automated tools available for the same purpose.
5. **Iterate:** Based on the feedback from **step 4**, steps 1 to 4 are repeated to incorporate changes in the ontology.

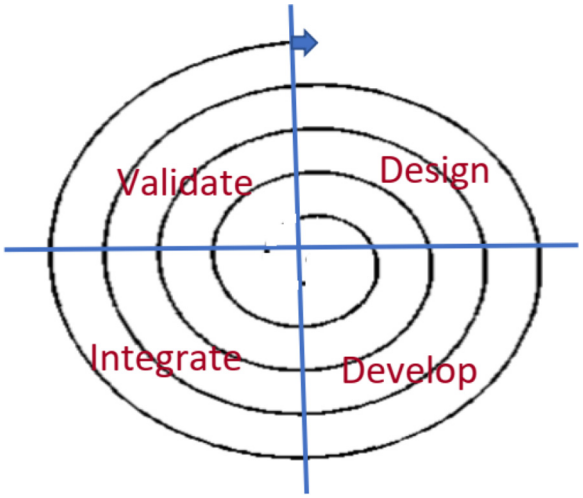

Fig. 3. Iterative process for ontology construction.

For the ontology development process itself, without any deviations, we choose to follow is the one that is proposed by Brusa et al., 2006 which is given in Fig. 5. The input to this process is the definition of the domain for which the ontology has to be developed. For instance, *ontology for the IoT ecosystem with SDDC.*

The detailed steps to be followed based on Brusa et al. (2006) are given by the authors in a practical consumable form in Noy and McGuinness (2001). We follow the similar approach.

1. Identification of the scope of the ontology

Before starting the creation of an ontology, the basic question one might ask is to establish the context and its scope. For this we ask several competency questions such as:

(i) What is the domain and what would it cover?
The domain of our ontology is the ecosystem of IoT deployed with SDDC.
(ii) What will be the use of our ontology?.
The ontology will be used to address the challenges of OT/IT convergence.
(iii) What are the types of queries the ontology must be able to handle and provide the responses? The following are some of the queries based on the problems to be solved by our ontology, but is not limited to:

(a) resource management queries such as:

- A security certificate accessible to IoT components by using the security certificate location (say cloud) and extending it to both IT and OT components.
- Querying a security policy applicable based on its device type.
- Inventory management queries to figure out all vendor model sensors, so that only such sensors are able to be updated by IT.
- Resource allocation and anticipating future needs.

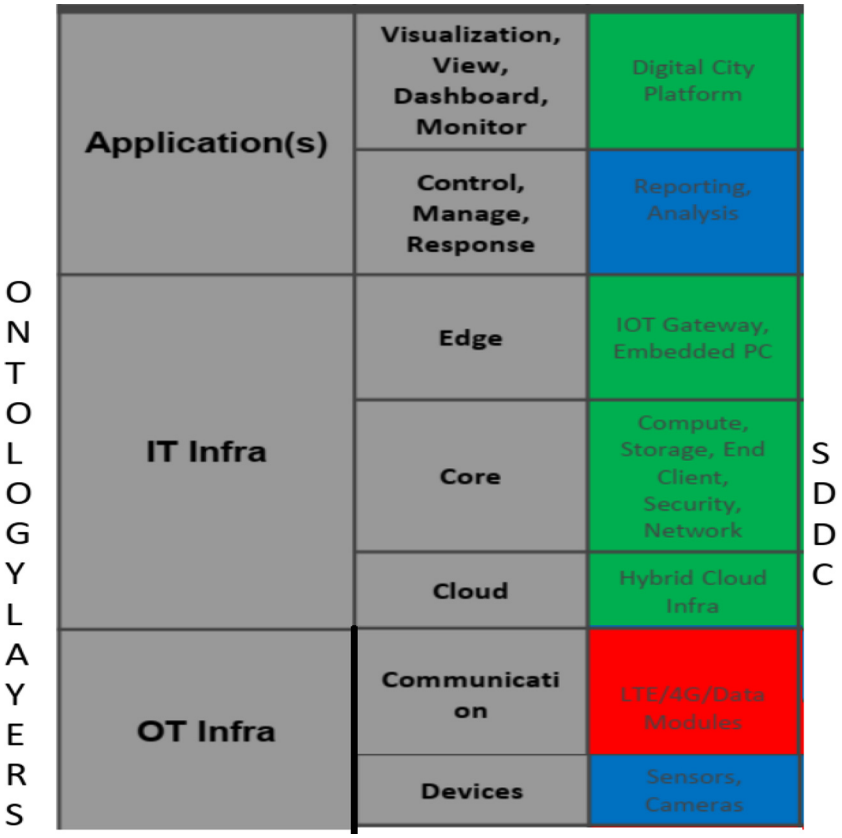

Fig. 4. Our proposed SDDC ecosystem unification of IoT ontology.

(b) Power profiling queries such as:

- Detecting sensor anomalies which would lead to power draining.
- Approximate consumption of power by IoT sensor/device, compute, network and storage from the IT side before the actual deployment.

(c) Queries related to failure of IoT devices/sensors such as:

- Details of all the sensors that are impacted due to their battery failures.
- Composite query to know the location of the sensors that have failed due to battery issue.
- What are the applications that are impacted due to the sensors failing?
- Lead time and the lag time for the operator to resolve the issue.
- List of all the sensors/devices from a given vendor that are showing symptoms of failures.

(iv) Who will be the consumers of the ontology and who will be subsequently maintaining the ontology?

The ontology will be consumed by smart infrastructure analytics applications which would be developed to monitor the IT/OT fabric for various use cases targeting the OT/IT convergence. The data center personnel involved in deploying IoT with SDDC such as the IT/OT admin, technicians etc., will be entrusted with the responsibility to maintain the ontology.

(2) Re-purpose of existing/available ontologies.

One of the main guiding principle to building new ontologies is not to reinvent the wheel and make use of any existing ontologies or parts of the ontologies. In our case, the Semantic Sensor Network (SSN) (Compton et al., 2012) ontology describing the sensors and their observations can be re-purposed by extending the ontology to include information/metadata related to the problems being solved for our IoT ecosystem. Likewise, the OpenIoT ontology (Soldatos et al., 2015) can also reused by making appropriate custom extensions to meet the requirements of our problem. To our best knowledge, we are not aware of an ontology for SDDC today. For the SDDC ontology, however, any of the current ontologies of cloud computing may be regarded as a reference and any common vocabularies/taxonomies between cloud and SDDC can be abstracted into a shared vocabulary.

(3) Ontology capture The main activities of this step are:

(a) Emphasize the essential ontological vocabulary/ taxonomy.
(b) Establish the hierarchy of classes and relationships.
We look at the taxonomies of each of the sub domains viz. IoT, Cloud computing and SDDC to gain an understanding of important vocabulary in our domain. Thus, for the IoT domain, we study the existing SSN ontology[3] as shown in Fig. 6 and understand its usefulness in our ontology construction.

[3] https://www.w3.org/2005/Incubator/ssn/ssnx/ssn.

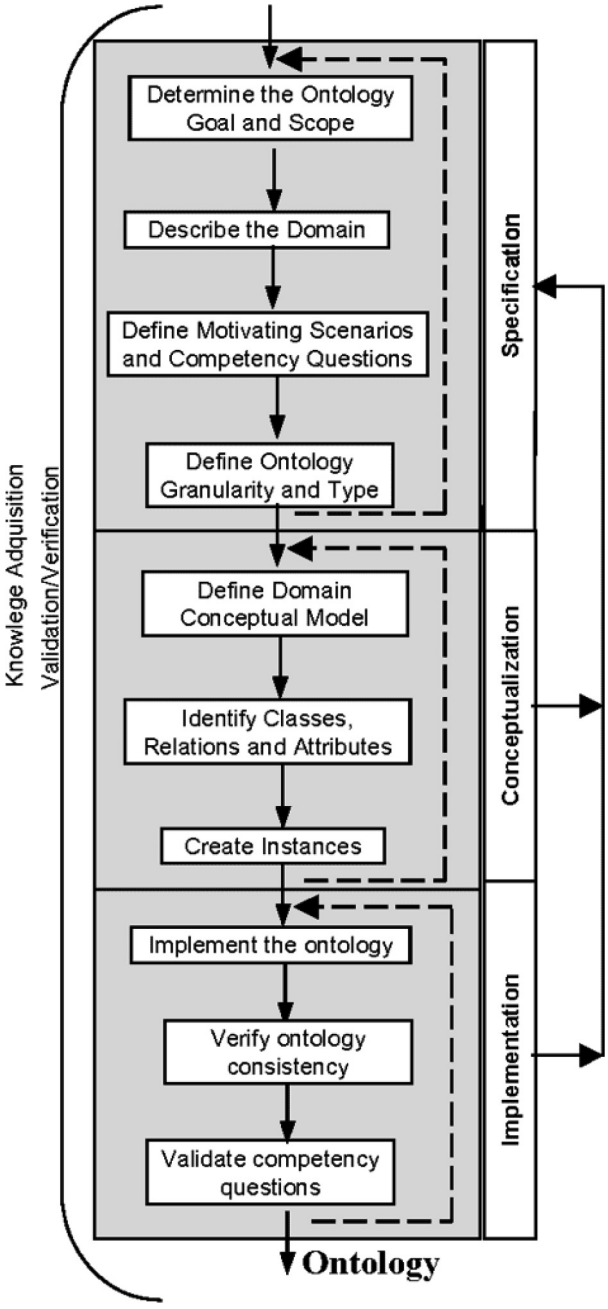

Fig. 5. Ontology development process.

Studying ontology for sensors (SSN), some of the vocabulary are, Device, Sensor, Observation Value, Survival Range and so on (Compton et al., 2012). The description for each can be found at on W3C.[4]

For cloud computing, we refer to the architecture for as proposed by National Institute of Standards and Technology (NIST) as shown in Fig. 7 in which we gather some of the important vocabulary for cloud are cloud infrastructure as a service (IaaS), virtualized resources (compute, network and storage) and so on (Bohn et al., 2011).

The taxonomy for SDDC is deduced based on the architecture from DMTF[5] (Distributed Management Task Force) as shown in Fig. 8. Some of the vocabulary are Virtualized resources, for instance storage, compute and network, Infrastructure as a Service (IaaS), security, infrastructure resource pools and so on.

4. Ontology encoding

After making sure of the availability of the taxonomies or vocabularies of the domain for which we need to model the ontology, the next task is to identify a language to represent our ontology is referred to as encoding. In the previous step, we have already identified the taxonomies for IoT, SDDC and Cloud. Whilst there are many implementation options such as OWL (Web Ontology Language), RDF (Resource Description Framework), we use the OWL2[6] from W3C to code our ontology based on our literature review(Kalibatiene and Vasilecas, 2011). We shall briefly summarize the properties of OWL. In the design and implementation of ontologies, there are four types of properties that play a significant role (Uschold et al., 2018):

(a) **Data Type property** which connects an individual to a literal. In OWL an individual refers to a specific thing. For instance, the house you live, John Doe's vehicle insurance policy and so on.
(b) **Object Type property** are used to relate two individuals.
(c) **Annotation property** to add description to the classes, individuals and object/data type properties. For instance, School [has] "An enclosed auditorium that is to host debates".
(d) **subPropertyOf** to represent a more specific kind of relationship than its parent property, For instance, employedBy is more specific than worksFor.

Before we proceed further, it is good to get an understanding of the operations that can be performed on ontologies.[7]

(a) **Merging of ontologies:** This operation involves the techniques used to arrive at a novel ontology by linking the existing individual ontologies. The knowledge is selected from the individual ontologies and imported into the new ontology, making sure that the end result is consistent.
(b) **Mapping of ontologies:** This operation is used to translate/map the concepts and relations from multiple ontologies. This may not always be possible and can leave behind inconsistencies in the end result.
(c) **Aligning of ontologies:** In this operation, new concepts and relations are defined to map the concepts in multiple ontologies that are participating in the alignment process.
(d) **Refinement of ontologies:** In this operation every concept of an ontology A maps to a concept in ontology B, such that the concepts in both the ontologies are equivalent.
(e) **Unification of ontologies:** In this operation, the concepts in one ontology are aligned to the concepts in another ontology so that there is a bidirectional inferencing between both the ontologies.
(f) **Integration of ontologies:** In this operation, the same portions of two different ontologies say O_1 and O_2 are considered to arrive at a new ontology O_3. This new ontology serves the purpose of translating between the two ontologies or sometimes can replace both the ontologies.
(g) **Inheritance of ontologies:** In this operation, an ontology O_1 inherits from an ontology O_2 the concepts, relationships, restrictions/axioms taking care that inconsistency is not left behind due to knowledge of O_1. Modular design of ontologies

[4] https://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628.
[5] https://www.dmtf.org/.
[6] https://www.w3.org/TR/owl2-overview/.
[7] https://www.obitko.com/tutorials/ontologies-semantic-web/operations-on-ontologies.html.

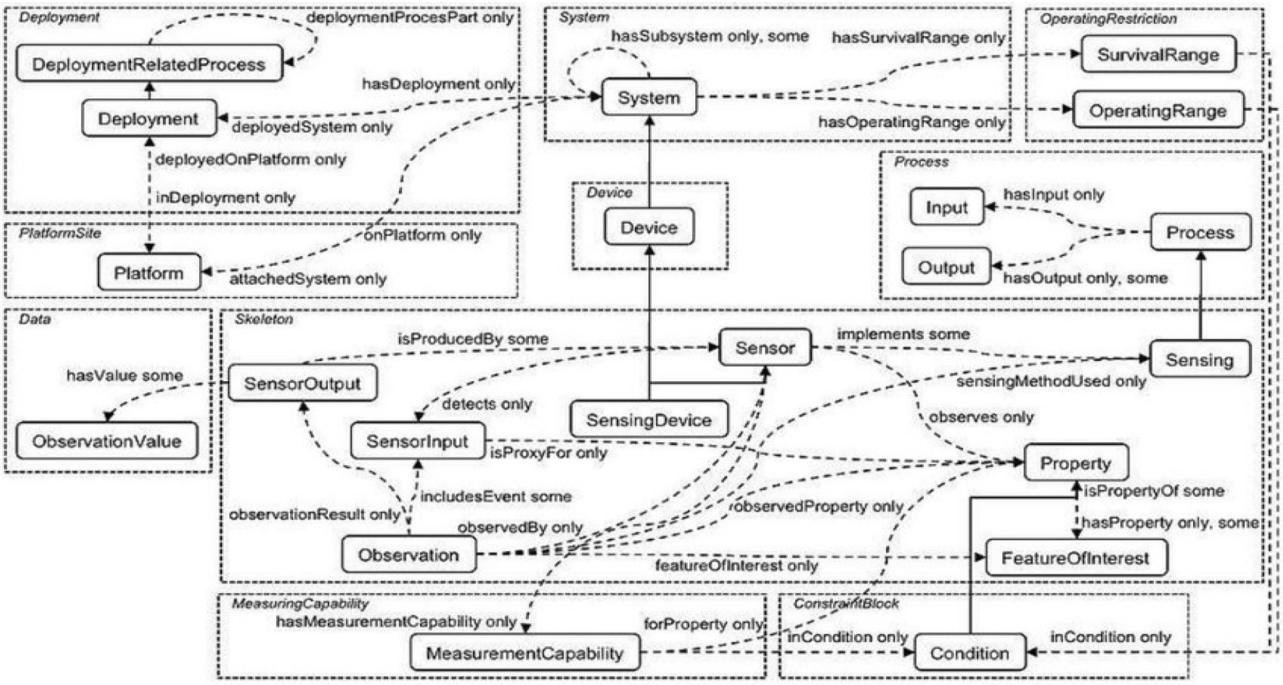

Fig. 6. SSN Ontology taxonomy from W3C (Compton et al., 2012).

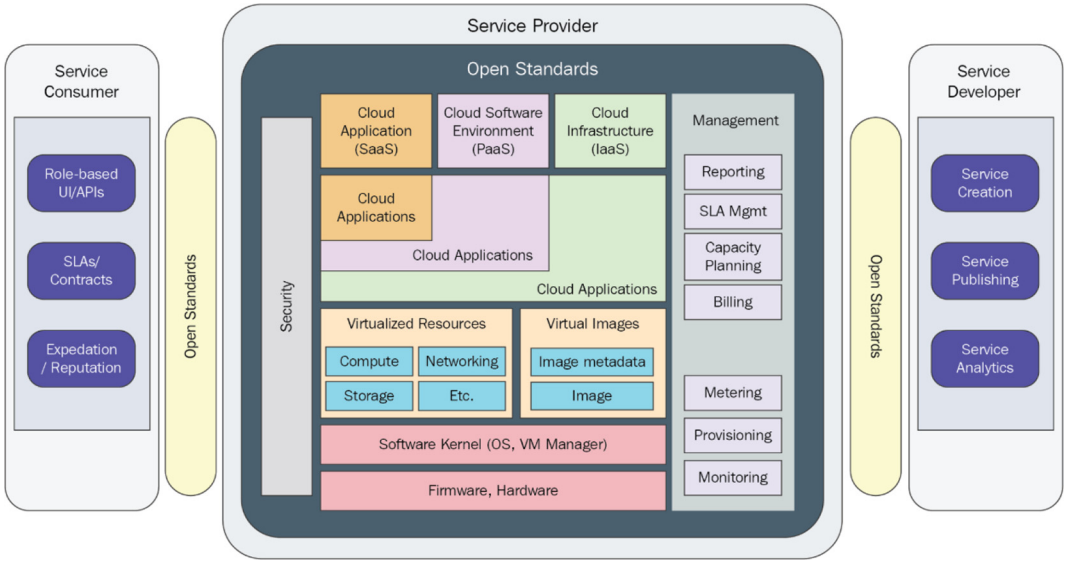

Fig. 7. NIST Cloud computing taxonomy (Bohn et al., 2011).

require inheritance where a high level ontology describing the knowledge expects the lower level ontology to only add additional knowledge required for the application.

Having gained an understanding of OWL and its constructs which is ontology language for encoding, and the operations that can be performed on ontologies, we need to identify an editor to start the implementation process. We select Protégé[8] (Musen, 2015) from Stanford University based on our literature review to compare the widely used ontology editors (Alatrish, 2014). We will now discuss each of the ontologies that form the individual ontology of SDDC's coherent ontology for IoT and the ontology operations which will lead to a unified SDDC ontology for IoT, which is our objective.

(a) **SDDC/Cloud Ontology**

For implementing the SDDC ontology we adopt the VMware Validated™ architecture[9] as shown in Fig. 9.
Our physical layer, in the real world represents the data center hardware which is compute, network and storage. Thus, we have a class *physical_layer* comprising of the sub classes *compute, network* and *storage*. We also get to access the ontology from the

[8] http://protege.stanford.edu.

[9] https://www.vmware.com/pdf/vmware-validated-design-30-sddc-reference-architecture.pdf.

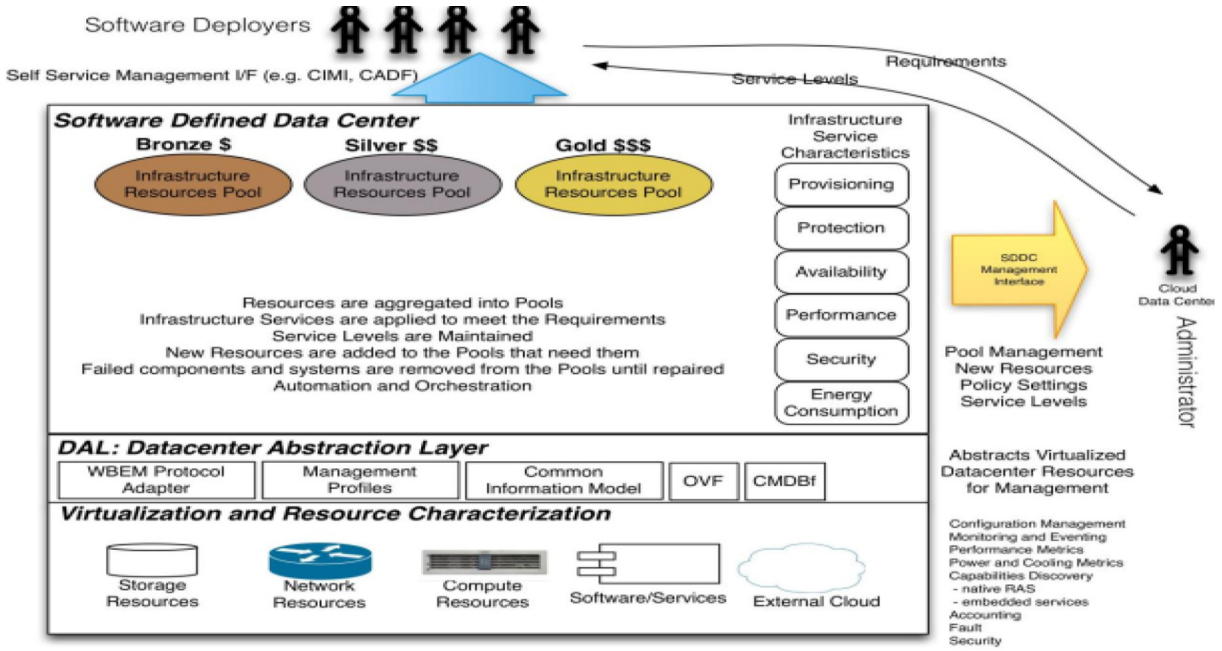

Fig. 8. SDDC Architecture and taxonomy.

authors of Memari et al. (2016) wherein the authors have designed a data center ontology for data centers for energy modeling. This ontology is useful to us to represent the *physical_layer* class. Thus, we will have a separate class *DCHardware* and its sub classes *Compute* (Servers with CPUs), *Storage* (of type Block Storage from Arrays, File Storage from NAS and Object Storage) and *Switches* (Network) to represent the Physical Layer infrastructure. The *hasDCHardware* object property associates the class *DCHardware* and class *physical_layer*. Generic properties of the class *DCHardware* which will be inherited by the sub classes as taken from our ontology development using the Protégé editor as shown in Fig. 10.

The next architectural element of SDDC is the Virtual Infrastructure Layer denoted by the class *virtual_infrastructure_layer* which consists of the mechanisms that abstract the elements of the physical layer. The main subclasses are *Hypervisor* representing software that abstracts/virtualizes the compute hardware to run Virtual Machines, *NetworkVirtualization* denoting **S**oftware **D**efined **N**etworking (SDN) aspects by segregating the control plane (sub class *ControlPlane*) from the data plane (sub class *DataPlane*) by managing the hardware switches of the physical layer. The attributes and properties of the *Hypervisor* class are captured in Fig. 11.

The controlPlane which manages the Switches and has the following attributes such as the flow rules for SDN, the vendor of the control plane software and others as shown in the Fig. 12.

An important architectural element of SDDC is the Cloud Management Layer. This is denoted as *cloud_management_layer* class and as a subclass of SDDC. Although Cloud Computing defines the layers Platform as a Service (PaaS), Software as a Service (SaaS) and Infrastructure as a Service (IaaS), for our research work, we are only concerned with IaaS. For this purpose, to check for the availability of any existing ontologies for cloud computing, we come across (Zhang et al., 2012) and we get access to this ontology called CoCoOn.owl. A thorough review of this ontology shows that we can reuse this ontology in our research to represent the cloud side of things. The major sub classes of this class are *AccessControl* (the credentials to access the cloud), *DeploymentModel* (Hybrid, Public, Private, Community), *Interface* (REST, WSDL), *ServiceLevelAgreement*, *Protocol* (CIFS, NFS etc). The major classes of the cloud_management_layer are as show in the Fig. 13. As we know, the cloud services representing the IaaS layer comprise of Compute, Network and Storage and these are very well represented in the cloud computing ontology.

(b) **IoT Ontology**

We start with the SSN ontology[10] as the base ontology and we also reuse the vocabulary from the IoT ontology from Kotis.[11] In doing so, for the *Device* class we introduce several classes to denote the various type of devices such as *Electrical* and *Mechanical* Device comprising of *Home Appliances*, *Electronic devices* comprising of the *Actuating Device, Computing Device* comprising of *Personal Computer, Processor* and *Server*. The IoT side of ontology with some of the important classes is as shown in Fig. 14.

5. Ontology integration and unification

The previous steps demonstrate how each ontology is designed and implemented i.e IoT, SDDC/cloud. The next logical step would be to unify these ontologies to represent our IoT ecosystem with SDDC which has the IT and OT sides of the real world. We introduce the class *ITInfra* representing the IT side which is essentially the SDDC/Cloud and we have the class *OTInfra* representing the OT side which is the IoT i.e., sensors/devices. The *DesignedArtifact* class of IoT ontology (SSN ontology) is linked to the *OTInfra* class and the *SDDC* class is linked to the *ITInfra* class. The unified ontology thus emerges as shown in Fig. 15.

6. Ontology Evaluation

To evaluate our ontology, we survey many tools and approaches. We choose to call our ontology as a large ontology since it represents the world of IoT ecosystem with SDDC achieved by the unification of the ontologies for IoT and SDDC/Cloud. The criteria to be considered for evaluating ontologies are Poveda-Villalón et al. (2014).

[10] https://www.w3.org/2005/Incubator/ssn/ssnx/ssn.
[11] http://ai-lab-webserver.aegean.gr/kotis/ontologies/IoT-ontology.

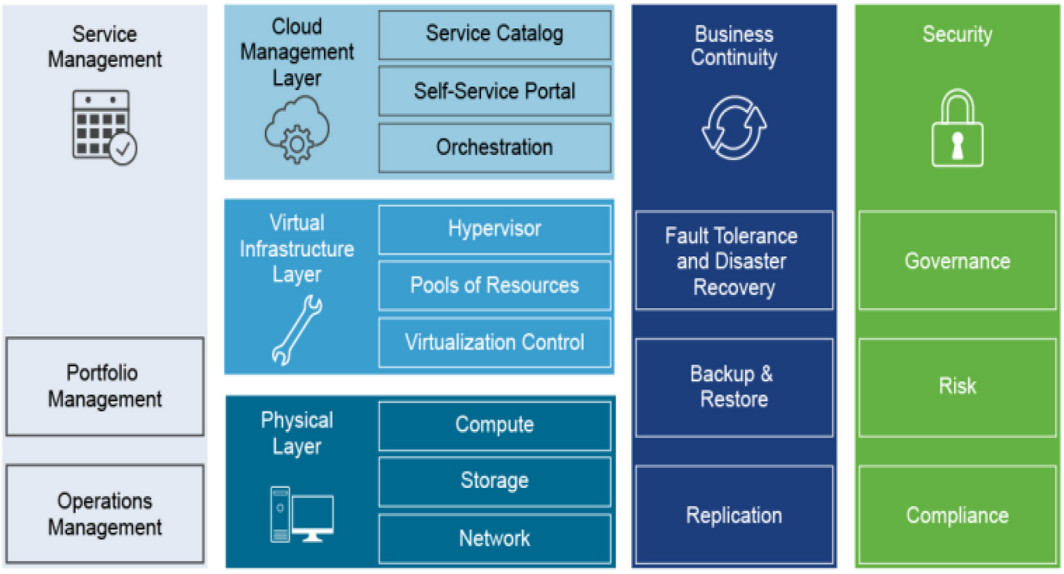

Fig. 9. VMware validated design for SDDC.

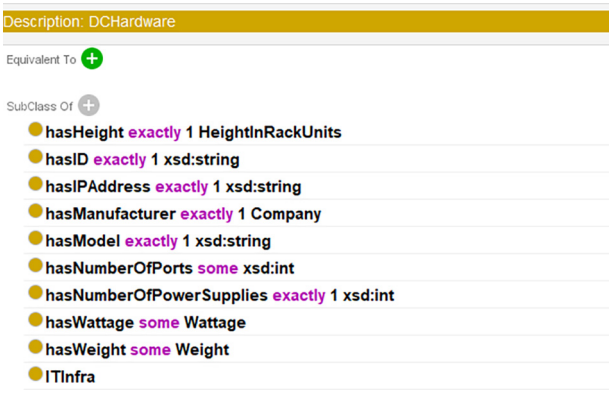

Fig. 10. Data center hardware class (DCHardware) and its associations.

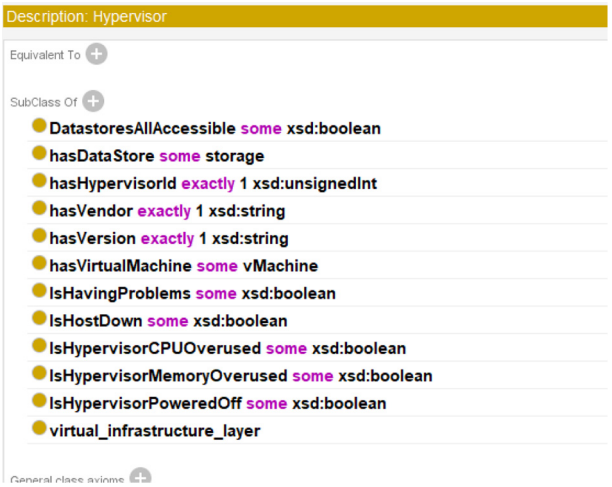

Fig. 11. Data Center Hypervisor class and its associations.

a) **Accuracy** relates to the correctness of the definitions, descriptions of classes, properties, and individuals in an ontology.
b) **Completeness** relates to coverage of domain of interest in the ontology.
c) **Conciseness** is about assessing if the ontology has any irrelevant elements in the covered domain of interest.
d) **Adaptability** looks at how strong is the ontology in laying a conceptual foundation for any range of tasks that are anticipated.
e) **Clarity** focuses on the effectiveness of the ontology in conveying the meaning of the defined vocabulary.
f) **Computational** efficiency looks at the tools and their ability to work with the ontology. For instance, the speed at which the reasoners need to satisfy the required tasks.
g) **Consistency** looks for any contradictions that the ontology might possess.

We come across the Ontology Pitfall Scanner (Raad and Cruz, 2015) (OOPS) which is an online resource to evaluate OWL/RDF schema. The scanner evaluates the supplied ontology resource either in the form of a URL or direct input as shown in the Fig. 16.

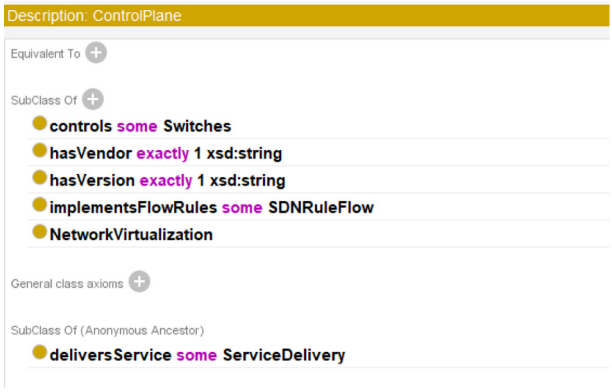

Fig. 12. SDN control plane attributes.

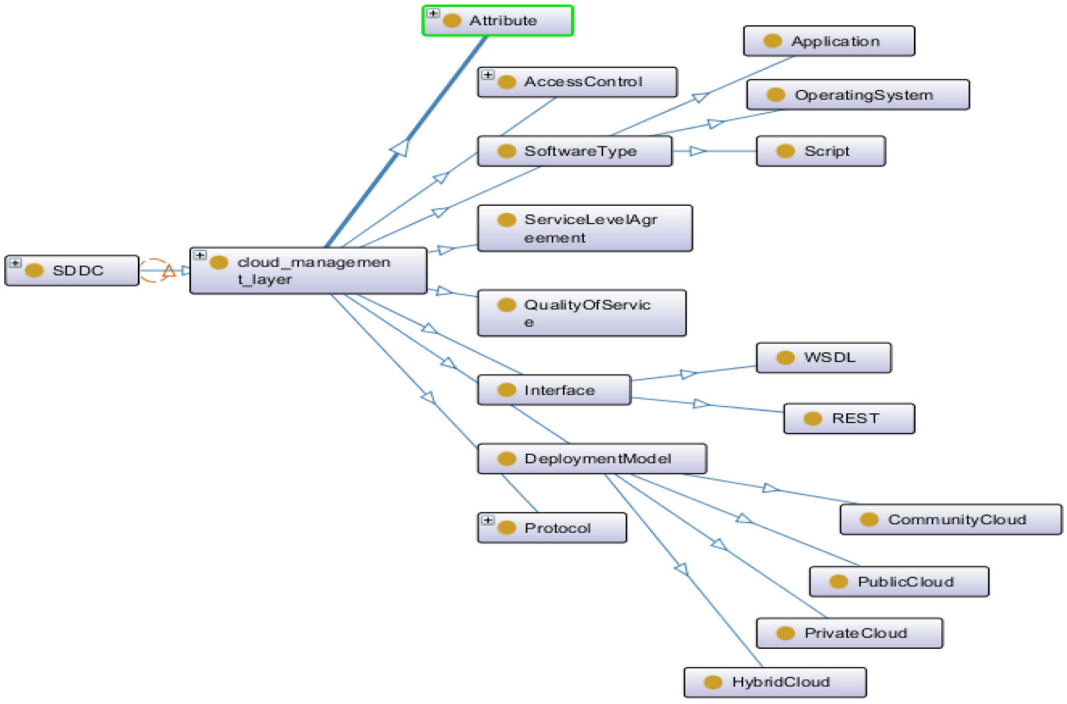

Fig. 13. Cloud management layer of SDDC.

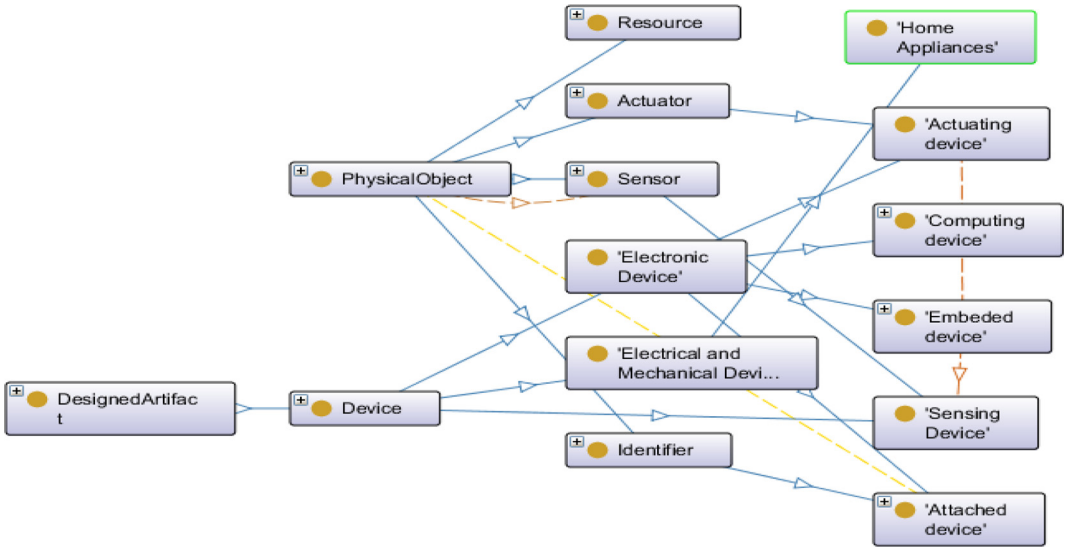

Fig. 14. Main classes of IoT ontology.

OOPS gives the users a choice of more than 40 pitfalls to select for evaluation. The pitfalls are labeled as P1, P2 and so on. Additionally, OOPS also provides the facility to evaluate ontologies based on categories such as Consistency, Completeness and Conciseness where for each category a subset of the pitfalls are evaluated. The pitfalls are described in Raad and Cruz (2015) which are based on categories and are documented in a catalogue.[12]

We iterate through the process of correcting the reported pitfalls from OOPs and rerunning the pitfall scanner many times. Towards that end, we reach a point where certain pitfalls are required to be corrected as per OOPS. As we are building a prototype and the ontology which would still evolve in the future research work where we will be adding more contextual information to the ontology to solve various research problems, OOPS will be rerun again many more times. A screen shot showing the evaluation of our ontology as output from OOPS is shown in the Fig. 17.

7. Ontology Documentation

The last phase of the design and implementation of an ontology is documentation. Information sharing where the constructed schema can be shared with others for reuse is one of the goals of ontologies, it is very important to document the ontology. For this purpose, we use the editors annotation property to document each class our ontology.

[12] http://oops.linkeddata.es/catalogue.jsp.

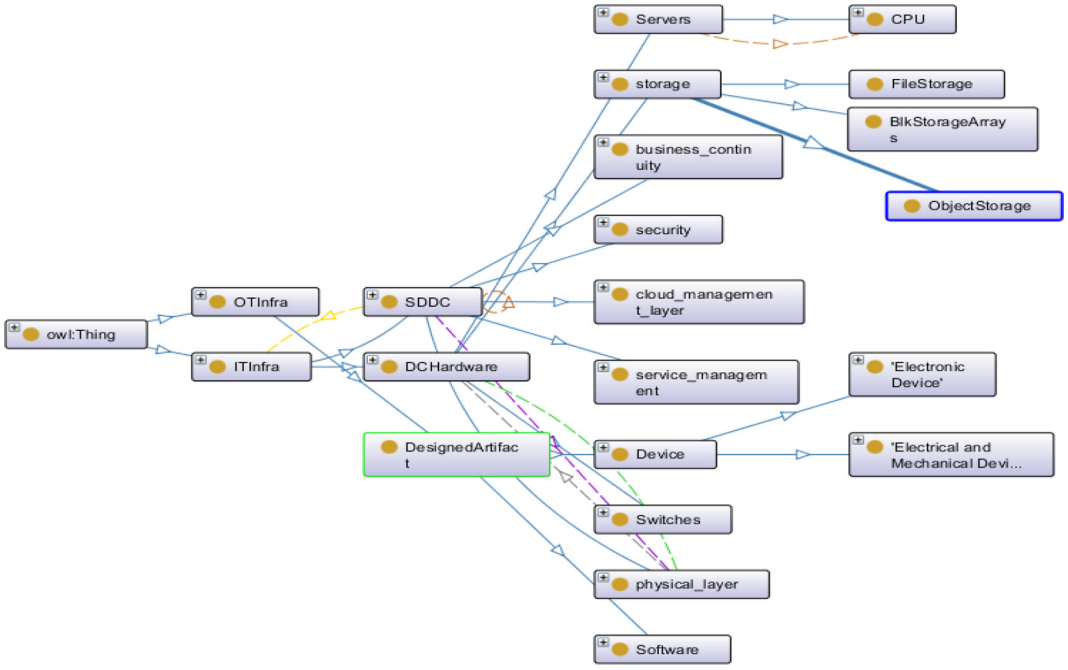

Fig. 15. Unified ontology showing IT and OT integration.

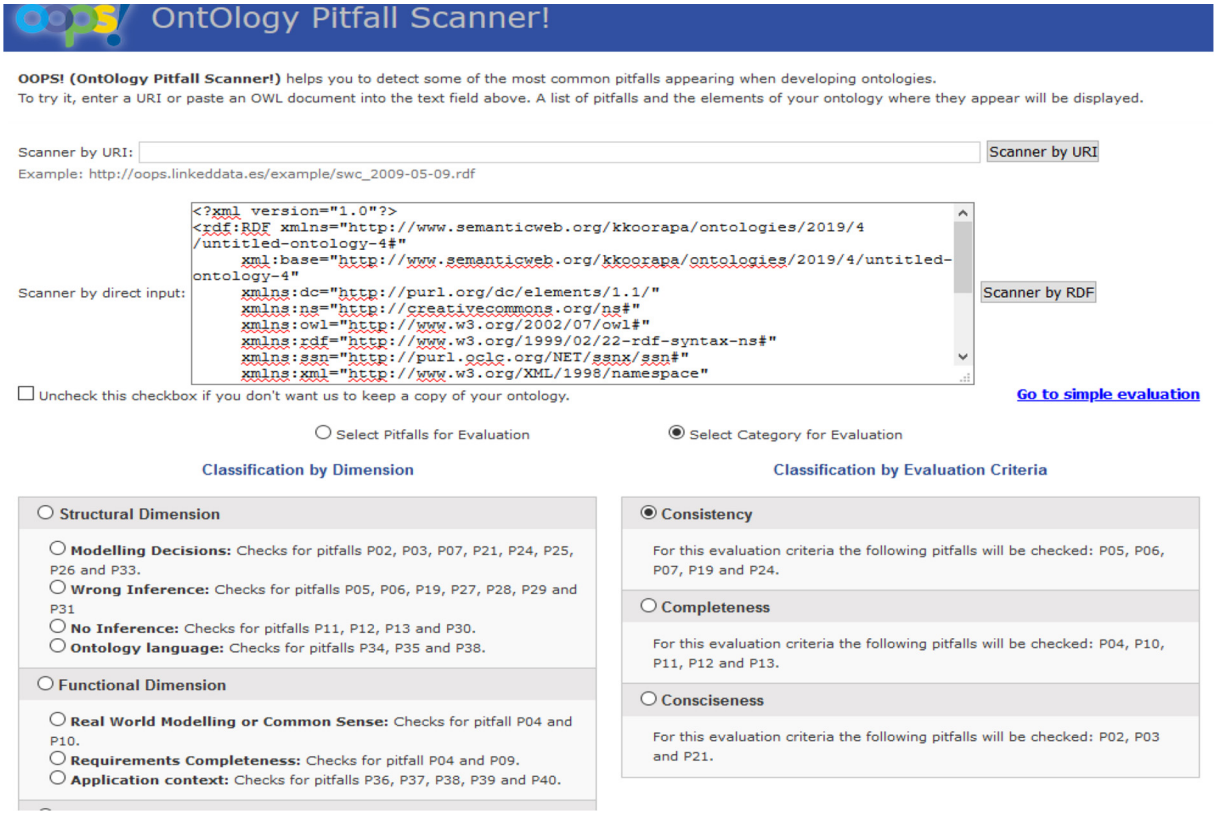

Fig. 16. Screen of ontology evaluation using OOPS.

6. Validation of use cases

With the unified ontology constructed so far, we now make use of it to develop smart applications for solving various use cases. We use Python based Owlready2[13] to load our ontology and write a test application. We come up with a rudimentary framework as shown in

[13] https://pythonhosted.org/Owlready2/.

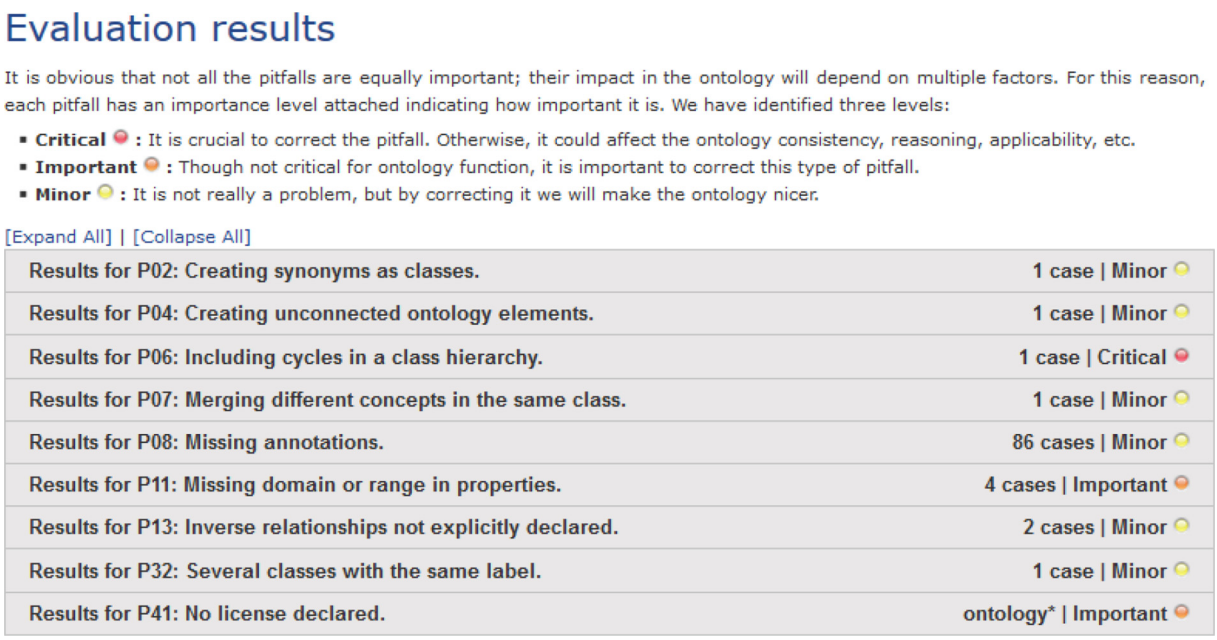

Fig. 17. Ontology evaluation output from OOPS.

Fig. 18 where the metadata from the IoT fabric can be sourced live from the fabric or through a file.

Using this framework, we now demonstrate the following use cases:

6.1. Device onboarding

The initial setup of an IoT device, after which the device becomes a part of the IoT fabric and starts its regular operation is known is very crucial which is what we refer to as onboarding of the device. This process as certain tasks which are required and are captured in the flowchart show in Fig. 19.

Thus, the whole onboarding process can be viewed as a function of the following:

1. Time to discover the device programmatically.
2. Time to validate the credentials manually in a siloed environment and automatically in a converged environment.
3. Time to provision the device in the network.
4. Time to validate that the provisioning actually works i.e verification.
5. Time to offload the configuration programmatically to device so the device can be bootstrapped to our OT network.

Even though ideally, scanning of RFID of devices is part of the device onboarding, in this work we do not consider this and assume that the device is already a part of the ontology. If each of the above is regarded as an operation and each operation taking a certain amount of time T_i $(1 \leqslant i \leqslant m)$, where m is the time required in seconds to complete the operation. If there are n activities/operations for each device, then the total time taken $T_{onboard}$ for onboarding num number or devices can be given as:

$$T_{onboard} = num * \sum_{i=1}^{n} T_i \tag{1}$$

Given the programmatic consideration of the devices, we assume that the standard worst case discovery time for the device is between 2 to 30 s and depending on how smart the device is, the credential validation is verified in 2 to 3 s (Associates, 2017). The graph in Fig. 20 shows the improvement achieved for onboarding of devices in an automated converged OT/IT IoT ecosystem.

Onboarding of the device can be carried out by the IT in several ways. Similarly, on-boarding can be incorporated by ontology as well. Since ontology offers a way to incorporate new devices and their schema and we have an ontology that provides the entire map of the entire end-to-end IoT ecosystem. We may also use the same security policy for a cell phone onboarding which we would use for a sensor onboarding. In the absence of an unified ontology, we must create an external bridge to bridge the IT-OT divide. Thanks to our unified ontology, the onboarding task can be carried out automatically and seamlessly.

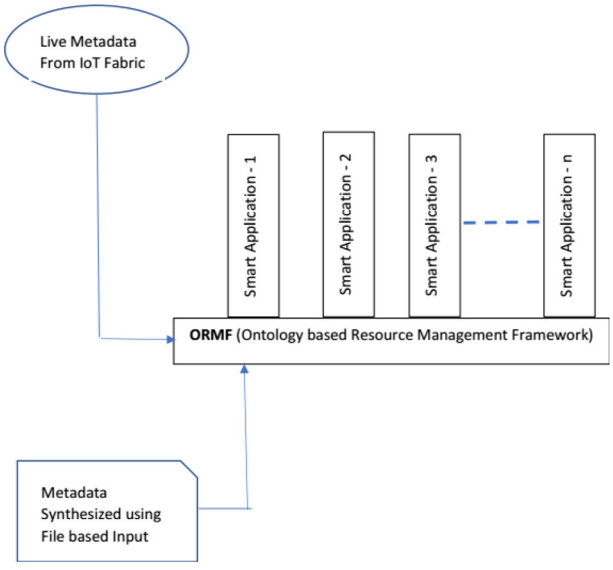

Fig. 18. Initial resource management framework using unified ontology.

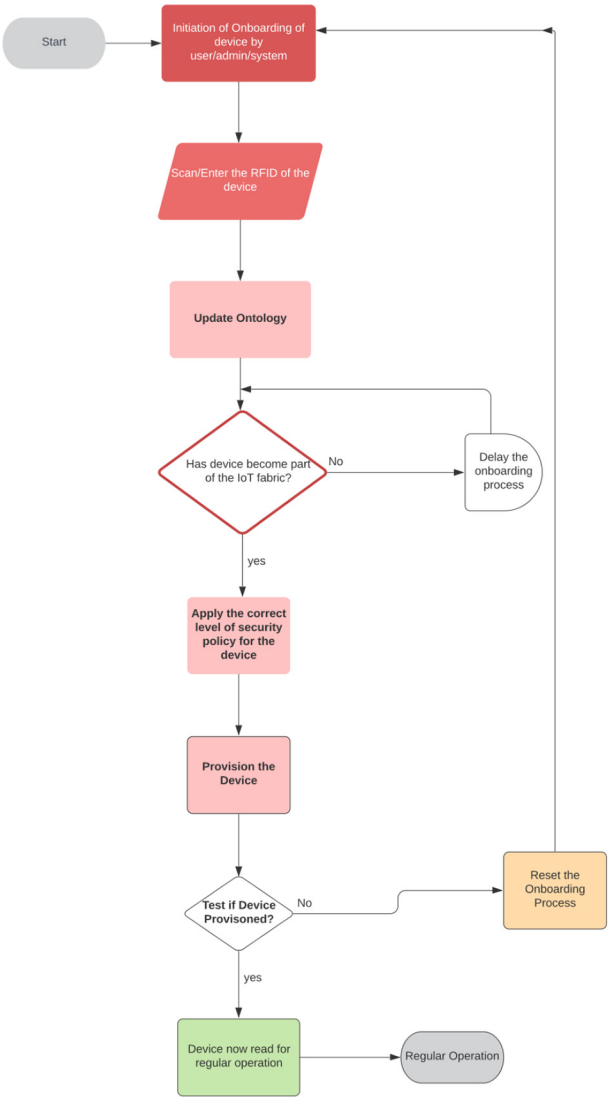

Fig. 19. Flowchart showing the steps involved in device onboarding.

6.2. Converged security

Another major advantage of a converged OT/IT is also a converge security approach over a siloed one is that the IT and OT security can be viewed in a single pane of glass. This means that the tools which are used by the IT to protect the data center, the same set of tools can be used to protect the IoT end points on the OT side. Otherwise, in a siloed environment two different sets of tools need to be used for OT and IT which will also mean two different sets of reporting. In a typical security operational aspect, we can consider the following parameters for a device:

1. Time to deploy/provision the device. This is the process of enrolling the IoT device into the IoT fabric. One of the important part of this process is authentication where a device presenting the right credentials is registered. Once a device plugs itself into the IoT fabric, it broadcasts its vital information such as its model and serial number to receive its other configuration data.
2. Time to secure the device. After a device broadcasts its vital data, as a part of securing the device, the following are of the common measures:

- Using a strong encryption method for Wi-Fi.
- Changing the default user names and passwords.
- Using strong, unique passwords for Wi-Fi networks and device accounts.
- Update the device to have the latest software patches.
- Install the right set of security policy.

3. Time to detect a security breach with the device.
4. Time to root cause the security breach with the device.
5. Time to isolate/quarantine the device.

The spider graph of Fig. 21 compares the operational security parameters of a siloed versus converged OT/IT IoT ecosystem. As we can see, in a siloed environment the activities such as detecting a security breach with a device, time to root cause the issue etc., are significantly higher when compared to the same in the converged environment.

7. Conclusions and future work

We now summarize our contributions to this research work and highlight our plans for future research in this area. We have seen

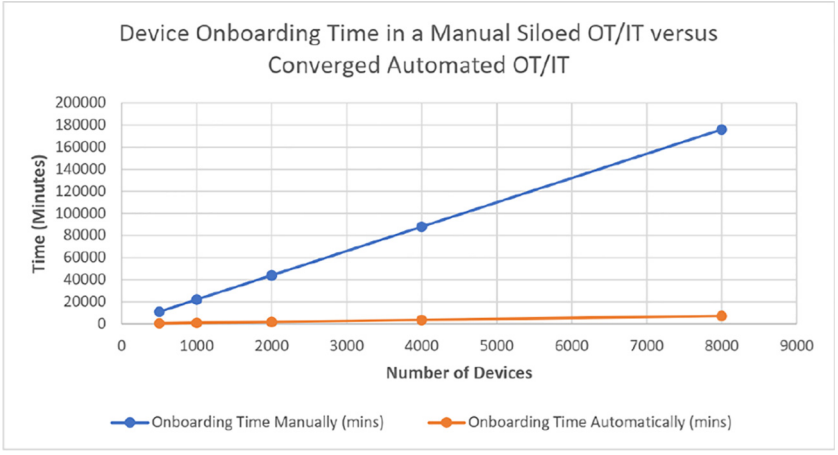

Fig. 20. Device onboarding times in a manual siloed OT/IT (without proposal) versus converged automated OT/IT (with proposal).

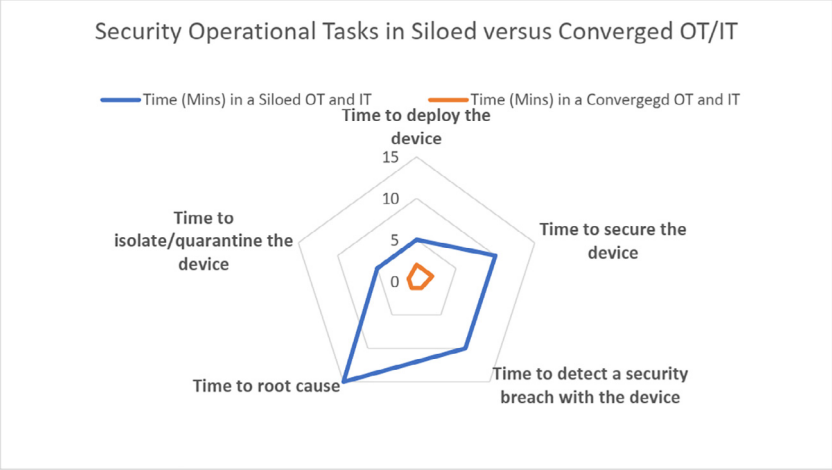

Fig. 21. Graph comparing operational security parameters in siloed (without proposal) versus converged OT/IT ecosystem (with proposal).

how to use a semantic-based approach like ontologies to holistically represent the IoT ecosystem with SDDC end-to-end and how to unify the ontologies of IoT, SDDC and Cloud using the concepts of ontology design principles to achieve coherent ontology. If the ontology unification did not happen, we would have constructed a collection of external applications that bridge the IT-OT gap. As we know, SDDC deals with operations concerning IT and thus infrastructure management, while the OT deals with the onboarding of sensors/devices and thus, sensor management. So, we are far more smoothly putting together IT and OT by uniting, thereby marrying proprietary OT with general IT. To achieve the logical connection of both the OT and IT sides which is very customized we need a broad range supervisory applications. We increase the degree of interoperability by describing them as an ontology. The applications do not need to worry about the sensors as long as they comprehend the ontology since the information about the sensors is abstracted. That is what our unified ontology has at its heart. We extend this further by proving that in order to combine IT with OT, we do not need to construct a personalized application(s), and applications should apply with ontology that is cohesive with IT and OT.

Some of the areas for future research include the formal verification of the schema, developing smart context-aware infrastructure analytics applications for end-to-end resource management, power profiling, predictive failure analytics. We also want to exploit OWL's advanced features to explore how misconfigurations in data centers can be identified using OWL (and likely advanced features) and OWL-based reasoning. With regards to complex event processing, it is currently a challenging task to characterize, identify and react to globally important event patterns, as metadata is often gathered from sources across the entire IoT detection ecosystem. Another interesting area is end-to-end SLA management for IoT fabric with SDDC. The IoT SLA (Service Level Agreement), thanks to its capability to identify and track quality of service (QoS) services, has received increasing attention. SLA is created by the negotiation process, whereby a subsection of clauses among options specified in advance by the provider is selected in the IoT application on the grounds of "take or leave it." The variability between IoT application and provider offerings could, however, lead to inappropriate SLA. The SDDC climate can also change over time and thus the SLA is unsatisfactory, which results in an alteration of the application requirement. Therefore, the SLA document can be generated with semanticized mapping of the IoT application requests and the provider. Additionally, we may include the need to adjust to the SLA through reasoning techniques and to re-negotiate automatically.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alatrish, E.S., 2014. Comparison Some of Ontology Editors..

Associates, K., 2017. IoT Onboarding – A Device Manufacturer's Perspective. Technical Report. Intel. URL:https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/kaiser-associates-iot-onboarding-for-device-manufacturers-whitepaper.pdf.

Bohn, R., Messina, J., Liu, F., Tong, J., Mao, J., 2011. Nist cloud computing reference architecture, pp. 594–596. doi: 10.1109/SERVICES.2011.105.

Chapple, M., 2015. IT-OT Security Convergence for Dummies. John Wiley and Sons Inc.

Brusa, G., Caliusco, M., Chiotti, O., 2006. A process for building a domain ontology: an experience in developing a government budgetary ontology 72.

Christy, P., 2017. When IT and Operational Technology Converge. Technical Report. Gartner Inc.

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Le Phuoc, D., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K., 2012. The ssn ontology of the w3c semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web 17, 25–32.

Cristani, M., Demrozi, F., Tomazzoli, C., 2018. Onto-plc: an ontology-driven methodology for converting plc industrial plants to iot, pp. 527–536. doi: 10.1016/j.procS.2018.07.287.

Delicato, F.C., Pires, P.F., Batista, T., 2017. Resource Management for Internet of Things. Springer Publishing Company, Incorporated.

Gruber, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum. Comput. Stud. 43, 907–928. https://doi.org/10.1006/ijhc.1995.1081. URL:https://doi.org/10.1006/ijhc.1995.1081.

Haase, P., Mathäß, T., Schmidt, M., Eberhart, A., Walther, U., 2010. Semantic technologies for enterprise cloud management, pp. 98–113. doi: 10.1007/978-3-642-17749-1_7.

Ismail, M.A., Yaacob, M., Kareem, S.A., 2006. Ontology construction: an overview. In: National Convension of Educational Technology.

Kalibatiene, D., Vasilecas, O., 2011. Survey on ontology languages. In: Grabis, J., Kirikova, M. (Eds.), Perspectives in Business Informatics Research. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp. 124–141.

Memari, A., Vornberger, J., Gómez, J.M., Nebel, W., 2016. A Data Center Simulation Framework Based on an Ontological Foundation. Springer International Publishing, Cham, pp. 39–57. https://doi.org/10.1007/978-3-319-23455-7_3. URL:https://doi.org/10.1007/978-3-319-23455-7_3.

Merritt, M.e.a., 2018. Protecting Industrial Control Systems And Critical Infrastructure From Attack. Technical Report. Forrester. URL:https://www.forrester.com/rep/Protecting+Industrial+Control+Systems+And+Critical+Infrastructure+From+Attack/-/E-RES139873.

Morliere, C., 2020. IT/OT Convergence, A fruitful integration of Information Systems and Operational Systems. Technical Report. Cigref. URL:https://www.cigref.fr/cigref-report-it-ot-convergence-a-fruitful-integration-of-information-systems-and-operational-systems.

Musen, M.A., 2015. The protégé project: a look back and a look forward. AI Matters 1, 4–12. https://doi.org/10.1145/2757001.2757003.

Olivieri, F., Cristani, M., Governatori, G., 2015. Compliant business processes with exclusive choices from agent specification. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9387, 603–612. https://doi.org/10.1007/978-3-319-25524-8_43.

Noy, N.F., McGuinness, D.L., 2001. Ontology development 101: a guide to creating your first ontology.

Olivieri, F., Governatori, G., Scannapieco, S., Cristani, M., 2013. Compliant business process design by declarative specifications. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8291 LNAI, 213–228. doi: 10.1007/978-3-642-44927-7_15.

Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C., 2014. OOPS! (OntOlogy Pitfall Scanner!): an on-line tool for ontology evaluation. Int. J. Semant. Web Inf. Syst. 10, 7–34. https://doi.org/10.4018/ijswis.2014040102. URL:https://doi.org/10.4018/ijswis.2014040102.

Raad, J., Cruz, C., 2015. A Survey on Ontology Evaluation Methods. SCITEPRESS – Science and Technology Publications, Lda, Lisbon, Portugal, pp. 179–186. https://doi.org/10.5220/0005591001790186. URL:https://doi.org/10.5220/0005591001790186.

Scannapieco, S., Governatori, G., Olivieri, F., Cristani, M., 2013. A methodology for plan revision under norm and outcome compliance. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8291 LNAI, 324–339. doi: 10.1007/978-3-642-44927-7_22.

Soldatos, J., Kefalakis, N., Hauswirth, M., Serrano, M., Calbimonte, J.P., Riahi, M., Aberer, K., Jayaraman, P.P., Zaslavsky, A., Zarko, I., Skorin-Kapov, L., Herzog, R., 2015. Openiot: open source internet-of-things in the cloud. Lect. Notes Comput. Sci. 9001, 13–25. https://doi.org/10.1007/978-3-319-16546-2_3.

Uschold, M., Ding, Y., Groth, P., 2018. Demystifying OWL for the Enterprise. Morgan & Claypool.

Youseff, L., Butrico, M., Silva, D., 2008. Toward a unified ontology of cloud computing, pp. 1–10. doi: 10.1109/GCE.2008.4738443.

Zhang, M., Ranjan, R., Haller, A., Georgakopoulos, D., Menzel, M., Nepal, S., 2012. An ontology-based system for cloud infrastructure services' discovery. IEEE, 524–530.