



HAL
open science

WoR Ontology: Modeling Resources in Web Connected Environments

Lara Kallab, Richard Chbeir, Sana Sellami, Omar Boucelma

► **To cite this version:**

Lara Kallab, Richard Chbeir, Sana Sellami, Omar Boucelma. WoR Ontology: Modeling Resources in Web Connected Environments. 2022 IEEE International Conference on Web Services (ICWS), Jul 2022, Barcelona, France. pp.286-295, 10.1109/ICWS55610.2022.00050 . hal-03782626

HAL Id: hal-03782626

<https://hal.science/hal-03782626>

Submitted on 21 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

WoR Ontology: Modeling Resources in Web Connected Environments

Lara Kallab

Open Group

R&D Department

Levallois-Perret, France

lara.kallab@open-groupe.com

Richard Chbeir

Univ Pau & Pays Adour

E2S UPPA, LIUPPA, EA3000

Anglet, France

richard.chbeir@univ-pau.fr

Sana Sellami* and Omar Boucelma†

Aix Marseille Univ, Université de Toulon

CNRS, LIS

Marseille, France

*sana.sellami@univ-amu.fr, †omar.boucelma@univ-amu.fr

Abstract—The Web of Things (WoT) describes a set of standards by the World Wide Web Consortium (W3C) for the interoperability of different Internet of Things (IoT) platforms, and different application domains. Thus, it guarantees not only device-to-device interactions, but also, application-to-application communications, despite their platform heterogeneity. To identify and use provided services (also called resources) that are exposed by either the devices or the applications connected to a Web environment, describing them using an open, shared and dynamic knowledge representation is required, allowing them to interoperate on both syntactic and semantic levels. In this paper, we propose WoR, a Web of Resources ontology that provides a modular and a common vocabulary to describe Web resources. WoR can: (1) ease the discovery, the selection, and the composition of different kind of resources (exposed by connected Web devices or Web applications), (2) provide reasoning means to discover new information, and (3) allow future extensibility and adaptation to new domains needs. Experiments were made to evaluate our proposed WoR ontology, showing promising results on the effectiveness and the performance levels.

Index Terms—Web Connected Environment, Web of Things, Resource, Semantic Modeling, Ontology, Composition

I. INTRODUCTION

The Internet of Things (IoT) is a concept that encompasses a wide range of objects (e.g., smart devices), embedded with electronics, software and sensors, that are connected to the internet to collect and share data [1]. With the heterogeneity of objects, it became challenging to seamlessly build a communication that allows appropriate interoperability. The same challenge appears in application-to-application interactions, as applications may be developed on different platforms and based on various programming languages. As a way to tackle such challenge, the Web [2] has emerged as a technology that guarantees the communication between the different connected objects and applications, despite their heterogeneity. Since then, the Web has become a major medium of communication platform [3] that led to the emerging of the Web of Things [4] (WoT).

In the Web context, the functions of the objects and applications are provided via Web services, i.e., self-contained, self-describing, modular components that can be published, located, and invoked across the Web [5]. Web services' implementation can mainly be done by using either SOAP (Simple Object Access Protocol) protocol [6], or XML-RPC (Remote

Procedure Call) [7], or following REST principles [8]. SOAP is a standard messaging protocol used by Web services to exchange XML data, and XML-RPC relies on calling procedures rather than manipulating resources like in REST, which is an architectural style that includes principles used to design and develop Web services as resources. Nowadays, REST has become the most used solution for implementing Web services [9]. This is due to: (i) its simplicity and ease of use that make services integration cost-effective, (ii) its support for different data formats other than XML, such as JSON, which is a more lightweight format compared to XML, and (iii) its ability to support caching for better performance and scalability. Hence, more and more objects and applications provide their functions as RESTful Web services that follow the principles of the REST architectural style. However, these Web services should be defined in a way that allows them to be largely adopted.

In fact, it is well acknowledged that semantic Web technologies hold the potential of achieving data and platform interoperability [10]. This is done by describing and sharing knowledge on Web services using unambiguous and machine-understandable vocabularies that attribute the same meaning to the specified and exchanged data. One of the building blocks in semantic Web Technologies, is the "Ontologies" [11], which provide a common and a comprehensible vocabulary for publishing data through a formal explicit specification of a domain conceptualisation. They facilitate knowledge sharing between systems across different organizations, allowing interoperability on both syntactic and semantic levels.

Over the last decade, many models and particularly ontology-based ones [12][13][14][15] have been built to specify IoT/WoT domains, including their exposed services (functions). These models have been defined for different reasons, among them, to represent a specific domain knowledge (e.g., specifying the entities, relations, and activities involved in sensing data by smart devices) in a machine-understandable way, allowing things with their functions (or services) to be discovered, aggregated, and remotely accessible. However, they have some limitations. As such, most models [16][15][13] have been defined to mainly focus on describing the services of connected objects, without considering the ones that can be provided by Web applications. This reduces the expressive

power when one explores and queries the connected environment. Also, and despite the fact that the majority of the models (as in [17] and [18]) include important concepts to describe a service provided by an object (as object locations, their provided functions with their necessary inputs and outputs), they lack in considering other important aspects related to links (e.g., a link denoting that a service is similar to another one as they provide the same function). Service (semantic) links can be useful while composing services, where services are linked together to form new value-added ones [19], when no single service can answer certain demands. Moreover, most of the models do not include the description of composed services, which can be used later in other different scenarios. And if some models do include composed services descriptions, they are limited and do not follow the REST architecture style principles [20]. The latter criteria is important to be considered as REST-based Web services are lighter for IoT/WoT devices compared to SOAP-based Web services [21], which are considered by most of the current existing models. In addition, recently, the concept of virtual devices (e.g., virtual sensors [22]), has emerged, to aggregate capabilities of a set of IoT devices to derive new services. These new services, referred to as virtual services or resources in this paper, and which are considered as value-added services besides the composed services as they allow to simulate the behavior of services of the devices that can not really exist, are not considered in existing IoT/WoT models.

To address the aforementioned limitations, we propose in this paper, an ontology-based Web resources model, called WoR (standing for Web of Resources), for Web connected environments, in which objects and applications can provide and exchange data. WoR provides a common vocabulary that is used to describe exposed Web resources (REST-based Web services) by both objects and applications. It is able to: (i) store and integrate resource specifications related to heterogeneous objects and applications, (ii) ease the discovery, selection, and composition of the exposed resources, (iii) provide reasoning means to discover new information, and (vi) allow future extensibility and adaptation.

The rest of the paper is organized as follows. Section 2 presents a scenario to motivate the usability and applicability of our work. Section 3 presents the related work, and shows the originality of our model. Section 4 details the specifications of our proposed Web resources model. Section 5 evaluates the effectiveness and the performance of the solution. Finally, Section 6 summarizes the work and gives future directions.

II. MOTIVATING SCENARIO

We motivate our work with a scenario illustrated in OpenCEMS¹: a Web platform that provides solutions for energy data management in connected environments. The platform allows connecting objects (stationary or mobile) and Web applications, both exposing Web resources for either: collecting on-site data, preprocessing collected data, or analyzing

¹Connected Environment & Distributed Energy Data Management Solutions: <https://opencems.sigappfr.org/>

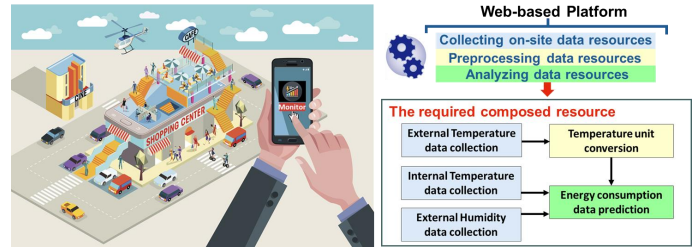


Fig. 1: An OpenCEMS instance applied in a smart shopping center, with an example of a composed resource

data. In the scenario, we consider a facility manager of a smart mall connecting a set of smart sensors (e.g., temperature sensors and humidity sensors) and Web applications (e.g., online weather applications), and providing several other Web resources (e.g., temperature and energy data predictions). The facility manager wants to predict the energy consumption of a specific floor of the mall, for the upcoming 2 days. The prediction results can help monitor the energy consumption of the corresponding floor, and allow establishing plans for energy supply and demand. Through a visualization interface provided by OpenCEMS, the facility manager can visualize the set of the resources provided by the mall and call the most convenient ones answering his need. To do so, several resources are to be selected and linked together to form the necessary composed resource, as shown in Figure 1: (i) data collection resources to collect the necessary data (e.g., external temperature and humidity data provided by an online weather application, and internal temperature data provided by installed sensors in the required floor), (ii) preprocessing data resources to prepare the collected data, if necessary (e.g., temperature unit conversion), and (iii) advanced data analysis resources to predict the energy consumption data. However, the existing of numerous and heterogeneous resources providing different functions, but also, in some cases, having the same required functions, along with their different characteristics in terms of response time, performance, etc., make resource identification and selection complex tasks for the facility manager. To facilitate these tasks, it is important to describe the exposed resources through a unified description model, that is expressed in a human-comprehensible and a machine-readable vocabulary. Such model should cover the following criteria:

- 1) **A thorough model.** Due to the various types of resources that can be connected to the shopping mall, it is important that the model allows their description to meet the different facility manager needs. Particularly:

- **Object and application resources:** As the shopping center can connect objects (e.g., smart sensors) and Web applications (e.g., online weather applications), it is important that the model allows the description of the resources exposed by these two different sources. In fact, the resources can be described using common concepts, such as the “Function” concept, which is a necessary to consider for all kind of resources (i.e.,

whether exposed by a temperature sensor or by an energy consumption prediction application), to know the exact function they provide. However, they can have, each, specific concepts, e.g., “Location” and “Operation Range”, which can only be assigned for the installed sensors.

- **Composed resources:** In some cases, there is no single resource that can satisfy a specific user demand. However, the combination of two or more resources, forming a composed resource (a composition), may provide the required outputs. This is the case of the facility manager demand that requires the use of several resources (for data collection, data preprocessing, and advanced data analysis). To enable the use of already formed composed resources and avoid repeating the composition process from the start, which consumes time and resources (CPU, memory, etc.), it becomes necessary to model and store the composed resources. The stored composed resources can be considered as single resources to be used in other scenarios.
 - **Virtual resources:** Some users demands may require to use data that cannot be collected by actual physical devices. For example, the facility manager may desire to collect temperature data from sensors that are not physically placed in the corresponding floor (e.g., due to high cost reasons). For such matters, virtual resources can be used to simulate the behavior of these physical sensors, by using some existing data and applying few calculations, in order to acquire the necessary results. Thus, allowing the description of virtual resources gets to be efficient for several cases.
- 2) **An expressive model.** Several description aspects should be taken into consideration to represent, at best, each provided resource, allowing its correct usage in different scenarios. Below we motivate the use of these aspects.
- **Provided function:** It is the first main aspect to consider for each resource, to help in identifying the suitable resources that can answer the facility manager need. For example, when requesting to predict energy consumption data, it is important to be able to identify the set of resources realizing such function.
 - **I/O:** Describing the Inputs/Outputs (I/O) of a resource allows to know the expected/returned data for/from each resource. As such, let's consider the phase when the facility manager wants to collect the necessary external temperature data surrounding the shopping center in C°, and which will be later used to predict the energy consumption data required for the corresponding floor. Several resources can be exposed to convert the collected temperature data to the necessary unit, but each, may require different inputs. To select the right one realizing user need, describing the necessary inputs required for each resource is important. Moreover, resources I/O help in ensuring that the resources are efficiently linked together while composing a resource.

For instance, there might be an I/O type mismatch between two linked resources (e.g., one with a float type data, and the other with a time-series type data), which can affect the quality of the predicted results.

- **Location:** In connected environments, like the smart shopping center, several devices are used for data collection. To collect relevant data and provide pertinent predicted results, it is crucial to precisely consider devices location. As such, when predicting the energy consumption of a specific floor of the shopping center, it is important to collect the necessary data (humidity, temperature, etc.) from the set of devices that are located in the corresponding floor.
 - **Links:** Defining links between the provided resources of an environment, can facilitate resource identification and selection, as well as ease resource replacement whenever a resource is no more available. For instance, when a resource is defined to be complementary to another (i.e., provides a function that is dependent of the other), the facility manager can easily know the possible resources to be chosen after selecting a specific resource. And when a resource is defined to be similar to another (i.e., provides a similar function), the facility manager can have several resource options (candidates) that satisfy his need. The similarity links between resources can also help replacing a unavailable resource (e.g., disconnected from the environment) by another similar one.
- 3) **A model supporting resource quality aspects.** When several connected resources provide the same needed function to answer the facility manager demand, it is pretty challenging to distinguish the most convenient one to his request, among the others. Therefore, it is important to differentiate between candidate resources having similar functions, by defining non-functional aspects, referred to as Quality of Service (QoS), which is in our case Quality of Resource (QoR). However, with the presence of resources that can be provided by either objects or applications, several QoR levels are to be considered:
- **Physical level:** refers to the attributes related to the physical devices (e.g., Operation Range and Battery). As such, the facility manager may prefer to collect temperature data using sensors with the highest operation range to acquire the most relevant collected ambient data.
 - **Network level:** designates the attributes related to the transferred data between the devices/the application (e.g., Bandwidth, Latency, etc.). For example, the facility manager may require fast results (e.g., in case of a fire detection), which can be affected by the amount of data transmitted between objects and applications. Thus, considering the Bandwidth or/and the Latency of the communicated data in the network is important.
 - **Application level:** represents the attributes related to the quality of services provided by each resource

exposed by a device or an application (e.g., Usage and Response Time). In this context, the facility manager may require using the resources that have been used many times, denoting a high usage rate. As such, the more a resource is called to answer user demands, the more it proves its efficiency in several scenarios.

In this paper, we embed the aforementioned criteria in a semantic Web resources model (ontology) explained in details in Section IV. The proposed ontology model is expressed using a vocabulary that is human-comprehensible to facilitate the identification, the selection, and composition of resources by end-users. Such vocabulary is also machine-readable that allows to automate all of these processes.

III. STATE OF THE ART

In this section, we study several IoT and WoT models, that were originally defined to enable interoperability between solutions from different providers and among various activity sectors in the IoT/WoT world, by semantically describing devices and data in order to make systems aware of their environment, its evolution, and the changes they can bring to it. While presenting their main core concepts/relations, we compare these models according to different criteria. The criteria, grouped into 3 categories (as shown below), are mainly related to the concepts and properties that are used to model the services exposed by the connected “Things” (e.g., Web objects and Web applications):

- 1) **Thorough model:** denoting the ability of the model to describe different types of services:
 - Objects and applications services, referring to the services that can be either exposed by connected objects, or connected applications
 - Elementary services, referring to the services that are not linked to any other service, and whose behavior is not simulated
 - Complex types of services, i.e., composed services and virtual services
 - Categories, referring to the services categories (e.g., data collection services and data preprocessing services), which can facilitate the exploration of services, and the understanding of their behavior
- 2) **Expressiveness:** indicating the ability of the model to cover various criteria representing the service:
 - Provided function
 - I/O, referring to the inputs and outputs of the services
 - Location, referring to the object location (whenever the services are exposed by connected objects)
 - Links, designating the links between the services provided by the connected objects/applications
- 3) **Resource quality:** designating the ability of the model to define resource/service qualities at various levels:
 - Physical, referring to the physical properties of the connected objects exposing services
 - Network, denoting the quality aspects of the data transfer/communication supported by the services

- Application, referring to the quality of the function provided by each service

A. IoT/WoT-based Models

Published as a W3C recommendation and as an OGC (Open Geospatial Consortium) implementation standard, SSN (Semantic Sensor Network) [23] and SOSA (Sensor, Observation, Sample, and Actuator) [24] are a set of ontologies that describes sensors, actuators, samplers as well as their observations, actuation, and sampling activities. These ontologies adopt a modular architecture with SOSA as a self-contained core that is extended by SSN and other modules to add expressivity and breadth. SSN and SOSA mainly cover the physical aspect of the IoT world (including sensors, actuators and samplers) and the modeling of the corresponding data. HSSN [25] is an ontology that extends SSN by adding concepts and properties related to sensor mobility, and multimedia data to cover: (i) sensor diversity, (ii) platform diversity, and (iii) data diversity. However, the definition of the interfaces and services (or operations) of the described IoT devices, is not included in any of these three mentioned models.

IoT-O [12], is a core-domain modular IoT ontology that proposes a vocabulary to describe connected devices and their relation with their environment. It includes five modules: (1) the Sensing module, which is based on SSN ontology [23] to describe the sensors and their observations, (2) the Acting module, which is based on SAN ontology² to describe how IoT devices can interact with the physical world (i.e., their performed actions), (3) the Lifecycle module that models state machines to specify IoT devices life cycle and usage, (4) the Service module, which describes the services provided by the IoT devices using MSM³, a REST architecture style [20]-based ontology, and (5) the Energy module that is defined by PowerOnt [26] to express power consumption profiles for appliances. DUL⁴ (DOLCE+DnS Ultralite) is an upper ontology that is used by these 5 modules. In the IoT-O model, any connected element can provide a Service, whether it is a physical object (ssn: Device) or an application (iot-o: Manager). Each Service is identifiable by an address, and can provide some Operation callable using HTTP method (e.g. GET and PUT). A method may have a set of inputs, outputs, and hypertexts to link the outputs of the operation to other operations. Although it is an ontology that describes REST services exposed by connected devices and applications, the IoT-O mainly lacks in considering composed/virtual services, their possible links, and QoS aspects related to Network and Applications levels.

oneM2M⁵ is a global standard for Machine to Machine Communications and the IoT, developed in an open and collaborative manner by many companies. oneM2M allows to annotate application specific resources (M2M data) with semantic description. It specifies a top-level Base ontology that allows

²<https://www.irit.fr/recherches/MELODI/ontologies/SAN.html>

³<https://lov.linkeddata.es/dataset/lov/vocabs/msm>

⁴<http://www.ontologydesignpatterns.org/ont/dul/DUL.owl>

⁵<https://www.onem2m.org/technical/onem2m-ontologies>

to create sub-classes (or equivalence classes) for application-level ontologies, e.g., Smart Appliances REference Ontology (SAREF)⁶. In the Base ontology of oneM2M, a Device has a Service that exposes some Operation. An Operation has some Operation Input and Operation Output, which are Variables that describe an Aspect, e.g., Temperature. oneM2M focuses more on machine-to-machine than Web applications, thus, it does not cover the services exposed by applications. In addition, it does not consider service composition or virtualization with the services possible links, nor even QoS aspects.

SAREF (Smart Appliances REference Ontology)⁷ is a modular and domain-independent semantic layer for smart appliances. It explicitly specifies recurring core concepts (e.g., Device, Function, Command, Service, Measurement, Property, and Unit of Measure) in the smart applications domain, the main relationships between these concepts, and axioms to constrain the usage of these concepts and relationships. In SAREF, a Service represents at least one Function and is offered by at least one Device that wants its function(s) to be discoverable, registerable and remotely controllable by other devices in the network. SAREF is a device-centric ontology that focuses on functions and measurements given by devices, neglecting application services, service composition and virtualization, and QoS aspects.

WoT TD (Thing Description) [17] is a formal model and a common representation for the Web of Things that is defined within W3C's WoT working group. A Thing Description describes the metadata and interfaces of Things, where a Thing is an abstraction of a physical or virtual entity that interacts and participates in the Web of Things. It provides a set of interactions based on a small vocabulary that makes it possible to integrate diverse devices and to allow diverse applications to interoperate. The WoT TD model is based on four main vocabularies: (1) the core TD Vocabulary, reflecting WoT's paradigm of Properties, Actions and Events, (2) the Data Schema Vocabulary, including the terms defined by JSON Schema (data types and data validation), (3) the WoT Security Vocabulary, reflecting the security mechanism and associated configuration requirements, and (4) the Hypermedia Controls Vocabulary, encoding the main principles of RESTful communication using Web links and forms. In the WoT TD model, Action is the function of the Thing, Property is used for sensing and controlling parameters, Form refers to the manner to access a function, and Links relates things together based on Web link specifications. The WoT TD model allows the description of objects and applications services, however, it does not include location devices, services I/O, service composition or virtualization, services possible links, nor even QoS aspects.

The IoT-A model [16] defines the main concepts of the domain IoT and describes the relationships between them. The concepts are the Entity, the Device, the Resource and the Service. Entity refers to "Thing" in the Internet of Things

and can be a human, animal or automobile object. The Device allows the entity to be part of the digital world by representing interactions. The resource represents an actual software component that provides information about the Entity or controls the Device. The Service provides a well-defined and standardized interface, offering the necessary functionalities to interact with entities. IoT-A therefore provides an architectural basis for other IoT-based projects. However, it does not include the services exposed by applications, service composition and virtualization with the services possible links, nor even QoS aspects.

Wang model [13] is a semantic description ontology for the representation of knowledge in the IoT domain. It consists on seven modules, namely: (1) the IoT Services module, which exposes the functionalities of the resources hosted on the devices, (2) the IoT Resources module that extends the SSN ontology by including other important resources in the IoT domain such as actuators, (3) the Observations and Measurements module, which allows to describe the real data generated, (4) the Physical Locations module, which includes concepts used for the lookup and discovery of IoT devices, (5) the Deployment Platform Networking module, which provides descriptions on how the IoT resources are organised and deployed as well as the system they form, (6) the Quality of Services and the Quality of Information module that includes important concepts used in many fields, in particularly in the composition and the adaptation of services for the providers and consumers of IoT services, (7) the IoT Service Test module that allows testing and verifying the functional and non-functional capabilities of IoT services during the design and deployment stages. The Wang model, however, lacks in considering the application provided services, composed/virtual services, and their possible links.

ForwardDS-IoT [15] is a semantic description model based on existing ontologies for the semantic description of objects using the SSN and SAN ontologies, their location by referring to the "Basic Geo" vocabulary⁸, which defines concepts such as the latitude, longitude and altitude of a geographic location, and IoT services using the OWL-S ontology⁹. However, ForwardDS-IoT lacks in covering the description of application services, virtual services, and services possible links.

SSN is an ontology that can be considered "too heavy" for a dynamic environment because of the large number of concepts it offers and which are often not used. IoT-Lite [27] is an instantiation of the SSN ontology to describe key IoT concepts enabling interoperability and sensory data discovery in heterogeneous IoT platforms through light semantics. The key concepts of the IoT-Lite model are related to three main classes: Object, System and Service, the latter being slightly described. As in SSN, IoT-Lite focuses on the physical aspect of the IoT world, without including the definition of the IoT devices' services (or operations).

⁶<https://saref.etsi.org/core/v3.1.1/>

⁷<https://saref.etsi.org/>

⁸<https://www.w3.org/2005/Incubator/geo/XGR-geo/>

⁹<https://www.w3.org/Submission/OWL-S/>

TABLE I: Comparative study of existing IoT/WoT-based models w.r.t. the identified criteria

	Thorough Model					Categories	Expressiveness				Service Quality		
	Objects Services	Applications Services	Elementary Services	Composed Services	Virtual Services		Provided Function	I/O	Location	Links	Physical	Network	Application
IoT-O	+	+	+	-	-	-	+	+	+	-	+	-	-
oneM2M	+	-	+	-	-	-	+	+	-	-	-	-	-
SAREF	+	-	+	-	-	-	+	-	-	-	-	-	-
WoT-TD	+	+	+	-	-	-	+	-	-	+	-	-	-
HSSN, SSN & SOSA	-	-	-	-	-	-	-	-	+	-	+	+	-
IoT-A	+	-	+	-	-	-	+	-	+	-	-	-	-
Wang Model	+	-	+	+	-	-	+	+	+	-	+	+	+
ForwarDS-IoT	+	-	+	+	-	-	+	+	+	-	+	+	-
IoT-Lite	+	-	+	-	-	-	+	-	+	-	+	+	-

B. Comparative Study

In table I, we show the comparative study of the existing IoT/WoT-based models previously described, according to the criteria presented in the beginning of Section III. We used the “+” symbol to express a positive coverage for a criterion, and the “-” symbol to express a lack of a criterion coverage. First, as seen in Table I, all of the existing models are not thorough. They mainly describe the elementary services, exposed only by devices. Few are the models that include composed services aspects within their descriptions (knowing that these models are SOAP-based and not REST-based), and none of the them consider virtual services that simulate the behavior of real ones. Second, all of the models include the functions provided by the services in their service description, and the majority describe services Input/Output parameters, as well as their location. However, they all lack in defining services links, which is an important criteria that can (1) facilitate service discovery and selection, and (2) ease service replacement whenever a service is no more available. This is apart of the WoT TD model [17], that although it supports services links (to know the services that can be called next to a current services state), the latters are not semantically defined. Third, although quality of services are covered by most of the IoT/WoT models, they are related to two main levels: Physical and Network, with a less attention to the Application level.

IV. WEB OF RESOURCES MODEL

In order to facilitate the identification, selection, and composition of RESTful-based Web services, i.e., exposed as resources either by WoT connected devices or by Web applications, and allow for their automatic process, we describe, in this section, a dedicated ontology called WoR (Web of Resources). WoR describes Web resources in a normalized way, through a vocabulary that can be used by different Web solutions/platforms.

A. WoR Ontology Features

WoR describes Web resources functional aspects (e.g., their provided functions, inputs/outputs parameters, etc.) and non-functional aspects (i.e., quality resources attributes related to their reliability, performance, security, etc.), including their composition features (whenever they are composed together in the same resource composition). It extends several ontologies:

- **HSSN**: As mentioned in the previous section, it is an extension of the SSN ontology, that models hybrid sensor

networks, i.e., networks containing mobile/stationary sensors, scalar/multimedia properties, and infrastructures/devices as platforms where sensors are deployed [25].

- **SOSA**: It is used to model the interaction between the entities involved in the acts of observation, actuation, and sampling [24].

Figure 2 shows the integration of these ontologies in WoR.

B. WoR Ontology Extensions

As shown in Figure 2, WoR extends existing model ontologies (HSSN and SOSA) by proposing new concepts and relations, while remaining compliant with existing standards. The extensions are related to:

- The thoroughness feature, by allowing resources to be exposed by devices and application platforms, and assigning categories to the resources. In our work, a resource (*WoR:Resource*) is defined as an RDFS¹⁰ resource type. This is inspired from Hydra vocabulary that is used to describe RESTful services by leveraging the power of Linked Data [28]. In addition, each WoR resource, can be defined either as an elementary resource (*WoR:ElementaryResource*), or a composite resource (*WoR:CompositeResource*), or a virtual resource (*WoR:VirtualResource*).
- The expressiveness feature, by defining reflexive semantic links between the resources, in addition to the resources provided function, defined in *WoR:Operation*, their I/O parameters, *WoR:Parameters*, and their location using *hssn:Location* (whenever they are exposed by connected devices);
- The resources quality aspects, by defining the *WoR:QoR* entity, which can be either *WoR:PhysicalQuality*, or *WoR:NetworkQuality*, or *WoR:ApplicationQuality*;

1) *Extensions related to the thoroughness resource model feature*: As shown in Section III, most of the IoT/WoT-based models focus on the resources that are only exposed by connected devices, without considering the resources published by Web applications. Therefore, and in order to allow the discovery, the selection, and the composition of all kind of resources, we extended WoR by adding the “Exposes” relation between each of the *hssn:Device* and *sosa:Platform* entities (with the latter being equivalent to the *WoR:Environment*), and *WoR:Resource* concept, denoting that a resource can be exposed either

¹⁰Resource Description Framework Schema: <https://www.w3.org/TR/rdf-schema/>

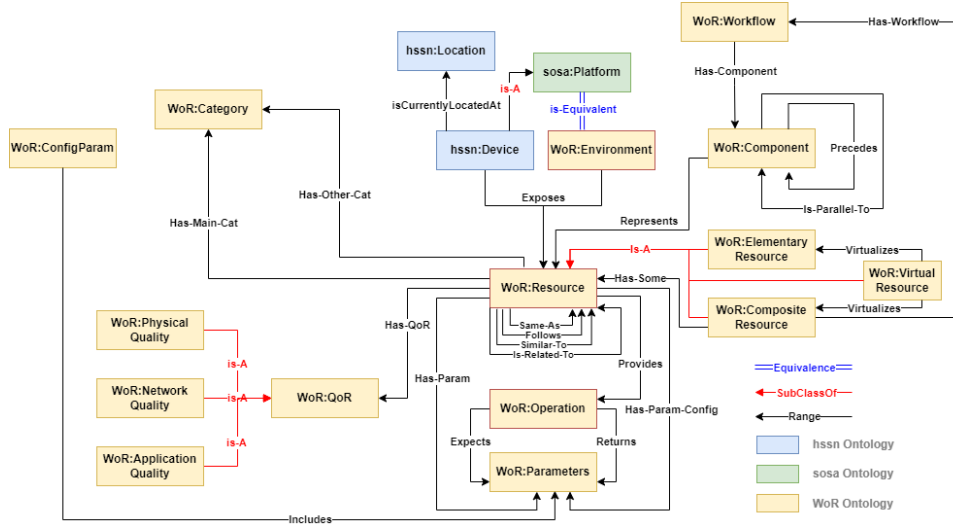


Fig. 2: Overview of the proposed Web resources model

by a device or by an application platform. Resources can be grouped into different categories, e.g., “Data collection”, “Data pre-processing”, “Advanced data analysis”, etc. Thus, each resource can be linked to a category (*WoR:Category*), allowing to organize the connected resources, and to facilitate the understanding of their Web related environment.

Moreover, in some cases, there is no single resource that can answer user request. However, combining two or more resources together in the same scenario, referred to as a resource composition, can generate the required results. To form a resource composition, resource discovery and selection are necessary, before being executed by a dedicated orchestration process [29]. Once formed, and in order to avoid re-executing both resource discovery and selection, which consumes several resources (e.g., CPU and memory) and time, it is important sometimes to store the composed resource and model it with the rest of elementary resources. For this aim, we distinguish in WoR between an elementary resource (*WoR:ElementaryResource*) and a composite resource (*WoR:CompositeResource*). The latter is formed by several other resources (composite or/and elementary ones). In order to allow the re-execution of a stored composed resource, we added the *WoR:Workflow* entity to detail the sequence process of the resources forming each composite resource. Such process is described by at least 2 components (*WoR:Component*) representing the resources included within the resource composition. The order of the resources is defined by the defined relations “Precedes” and “Is-Parallel-To”.

Also, in order to allow to mimic the behavior of missing real resources, e.g., the resources exposed by physical devices that are very costly to use, we allow in WoR to simulate such behavior, by defining virtual resources through the *WoR:VirtualResource* entity, a sub-class of the *WoR:Resource* that can virtualize both the *WoR:ElementaryResource* and the *WoR:CompositeResource* entities.

2) *Extensions related to the expressiveness resource model feature*: Hydra model, from which we were inspired to define a Web resource as an RDFS type resource, is a vocabulary that specifies a number of concepts that are used to describe resources, while enabling a server to advertise valid state transitions to a client. Within this context, Hydra defines the Link concept that enables the dynamic discovery of the next resources that can be called next at runtime. However, there are no semantic data about the type of the Link (e.g., whether it is complementary or similar for instance), which can facilitate the automatic resource discovery, selection, and composition, as well as the replacement of a resource in case it is no more available. Therefore, we define in WoR reflexive semantic relations between the resources:

- “Same-As” link, denotes that the related resources provide the exact same function
- “Follows” link, denotes that the related resources can be executed in a complementary manner based on the dependency of their provided function
- “Similar-To” link, indicates that the related resources provide a similar function
- “Is-Related-To” link, indicates that the related resources are exposed by devices installed in the same zone/location

The expressiveness model feature is also covered by defining the provided function attribute to the *WoR:Operation* of the exposed resources, with its input parameters (through the “Expects” relation between *WoR:Operation* and *WoR:Parameter*), its output parameters (through the “Returns” relation between *WoR:Operation* and *WoR:Parameter*), as well as resources location (*hssn:Location*) whenever they are exposed by devices. We note that in our model, the *WoR:Operation* represents the information necessary for clients to construct valid HTTP requests in order to call/manipulate the resource. As such, each *WoR:Operation* consists of a required HTTP method, optional input and output parameters, and information about the function provided by each resource Operation, such as

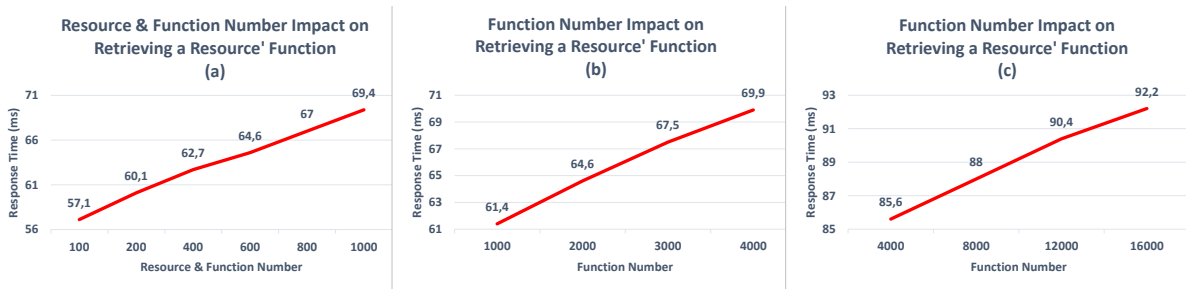


Fig. 3: Resource and Function number impact

“Collect temperature”, or “Predict energy consumption”, etc. Moreover, we defined the concept *WoR:ConfigParam* to represent the configurable coefficients (parameters) that are used to control or fit the Web resource models (whenever it is necessary) to data. An example of a configurable coefficient is the regularization parameter defined to a SVM (Support Vector Machine) model optimization, which serves as a degree of importance that is given to miss-classifications. The larger it is, the less the wrongly classified examples are allowed.

3) *Extensions related to resources quality aspects*: In many cases, there are several resources that can be discovered having the same or similar function. In order to distinguish between such resources, and select the most convenient ones answering user request, we defined, in WoR model, quality of resources (*WoR:QoR*), which can be divided into 3 main groups: (1) Physical quality (*WoR:PhysicalQuality*), representing the aspects that describe the quality of the IoT/WoT devices exposing the resources (e.g., Battery and Operation Range), (2) Network quality (*WoR:NetworkQuality*), denoting the aspects that specify the quality of the data transferred between the resources (e.g., Bandwidth and Latency) and (3) Application quality (*WoR:ApplicationQuality*), representing the quality of the service provided by the resources (e.g., Response Time).

V. EXPERIMENTAL EVALUATION

In this section, we present the experimental protocol that we followed to evaluate WoR ontology. It is based on 2 parts:

- 1) *Effectiveness Evaluation*: In which we verified if the concepts/properties defined in the ontology are able to meet the different objectives, presented in Section V-A below, and to cover the criteria explained in Section III.
- 2) *Performance Evaluation*: In which we studied the response time of different simple and complex queries, with the evolution of the resources graph based on several variations (e.g., increasing the number of devices exposing resources, and increasing the number of resources).

A. Effectiveness Evaluation

In this part, we define the most useful queries that cover our objectives in terms of: (i) Exploration, to search for the set of the resources that are connected, and to have a better understating of the Web environment, (ii) Discovery, to identify the necessary resources that respond to user demands, (iii) Selection, to select among the resources providing the same

required functions, the best ones answering user demands, and (iv) Composition/Execution, to link the identified and selected resources together, forming a composition, and allow for their execution according to their corresponding order. Moreover, we analyze these queries according to their ability in supporting the criteria presented in Section III.

In Table II, we present the set of the queries that we defined to test the effectiveness of our WoR model (in terms of covering the aforementioned specified objectives and necessary criteria). The queries are expressed in SPARQL¹¹ under the following link: https://github.com/AnonymousUser006/IEEEICWS2022/blob/main/WoR_AppendixA.pdf. In the same link, we also present the list of the SPARQL queries that are necessary to form the required composed resource related to the motivating scenario (see Section II).

B. Performance Evaluation

In order to evaluate the performance of the WoR ontology, we considered several scenarios to study the impact of the evolution of the resources’ graph on WoR’s performance. The performance is based on the execution of several queries (picked from the list of queries defined in Section V-A) in different scenarios consisting on different simulated resources’ graph settings: (1) varying the number of resources with their provided functions, (2) varying the number of devices exposing data collection resources and their distribution into different number of locations, (3) varying the number of involved resources in a composition. In the experiments, we show the query response time (ms) based on the average of 10 sequential executions for each query. The tests were conducted on a Windows 10 Professional machine with an Intel i7-8665U CPU @ 1.90GHz 2.11GHz processor and 1 GB RAM, using Stardog (<https://www.stardog.com/>), an Enterprise Knowledge Graph platform and graph DBMS with high availability, high performance reasoning, and virtualization.

1) *Resource and Function Impact*: In the first scenario (see Figure 3), we studied the impact of varying the number of exposed resources with the number of their provided functions. In Figure 3-(a), the number of resources is equivalent to the number of functions, where each resource provides one single function. In Figure 3-(b) and Figure 3-(c), we fixed the number

¹¹A standard query language and protocol for Linked Open Data on the Web, that is able to retrieve and manipulate data stored in Resource Description Framework (RDF) format.

TABLE II: List of useful queries covering the required objectives and criteria

	Query	Objectives				Criteria		
		Exploration	Discovery	Selection	Composition/ Execution	Thorough Model	Expressiveness	Resource Quality
A	Retrieve different types of Web resources	+	-	-	-	+	-	-
B	Retrieve the Web resources providing a given function	-	+	+	-	-	+	-
C	Retrieve the list of all the functions provided by the Web environment	+	-	-	-	-	+	-
D	Retrieve the output parameters of a Web resource, and the input parameters of another	-	-	+	-	-	+	-
E	Retrieve the Web resources exposed in a given location	+	+	+	-	-	+	-
F	Retrieve the Web resources that are the same (same-as) as a Web resource	-	+	+	-	-	+	-
G	Retrieve the Web resources that are complementary (follows) to a Web resource	-	+	+	-	-	+	-
H	Retrieve the workflow of a composed Web resource	-	-	-	+	+	-	-
I	Retrieve the sequential order of the workflow components of a composed Web resource	-	-	-	+	+	-	-
J	Retrieve the functions provided by the virtual Web resources	+	-	-	-	+	+	-
K	Retrieve a Web resource providing a given function with a quality criterion (e.g., Effectiveness>80%)	-	+	+	-	-	+	+
L	Retrieve a Web resource collecting data within a location with a quality criterion (e.g. Bandwidth>400 Mbits/sec)	-	+	+	-	-	+	+

of resources, respectively, on 500 and 4000, and increased the number of functions. In the latters, a resource provides more than one function. In each of these resources graph setups, we retrieved the list of resources that provide a specific given function by applying the query B (see Table II) and measured its corresponding response time. The increase of the query runtime in all of the graphs presented in Figure 3 is explained by the number of additional resources and functions expanding the search environment. However, we can see that the number of resources has more impact than the number of functions, as the response time is almost the same (64,4 ms and 64,9 ms) when having 1000 resources and 1000 functions (Figure 3-(a)) versus 500 resources and 4000 functions (Figure 3-(b)).

2) *Device and Location Impact*: In the second scenario (see Figure 4-(a) and Figure 4-(b)), we studied the impact of varying the number of devices exposing data collection resources and their distribution ratio in different numbers of locations. In Figure 4-(a), we increased the number of devices exposing Web resources and distributed them, equally (i.e., each location contained the exact same number of devices), on a fixed number of locations (50 locations). In these tests, the distribution ratio, which is equal to the number of devices divided by the number of locations, was thus evolving. In Figure 4-(b), we fixed the number of devices exposing resources (1000 devices) and distributed them, equally, on a varied number of locations. In such cases, the distribution ratio was decreasing. For the configuration sets in Figures 4-(a) and 4-(b), we retrieved the list of resources that are exposed in a specific given location by applying the query E (see Table II) and measured its corresponding response time. The increase of the query run-time in the graphs is explained by the number of additional devices and locations expanding the search environment. We can also observe that the evolving number of locations in which the devices are distributed has more influence than increasing the number of devices. This can be seen from the response time that has risen more (from 59,1 ms to 72,4 ms) when multiplying the number of locations

by 10 (from 50 to 500), comparing to multiplying the number of devices by 10 (from 100 to 1000) where the response time increased from 59,1 ms to 68,1 ms.

3) *Involved Resources in a Composition Workflow Impact*: In the third scenario (see Figure 4-(c)), we studied the impact of increasing the number of resources involved in a composition workflow. In this scenario, we retrieved the list of resources that form a composed Web resource by applying the query H (see Table II) and measured its corresponding response time. As shown in Figure 4-(c), the response time evolves almost linearly with the increasing number of resources, as there are more resources involved to get.

Discussion. The generated graphs in all of the different experimental scenarios show a promising and positive linear curve, denoting that the query response time increases linearly with the increasing number of resources, their provided functions, the number of connected devices (exposing resources) and their distribution into many locations, as well as the number of resources forming a resource composition. This indicates a proportional relation, which is a quasi-constant increase between the different variables used and the response time of the queries. The results also highlight that the growing number of resources and the distribution of the connected devices in many locations, has more impact on the increase of the queries response time, comparing to the other variables.

VI. CONCLUSION

In this paper, we propose a Web of Resources ontology (WoR), that is used to describe the services (the resources) provided by the devices and/or applications exposed by connected Web environments. It includes the functional and non-functional aspects of the resources, as well as their composition features (whenever they are composed together). WoR mainly reuses several existing known IoT-based models (i.e., HSSN, an extension of the Semantic Sensor Network ontology (SSN), which adds sensor mobility and multimedia data related concepts, and SOSA (Sensor, Observation, Sample, and Actuator ontology), which describes sensors,

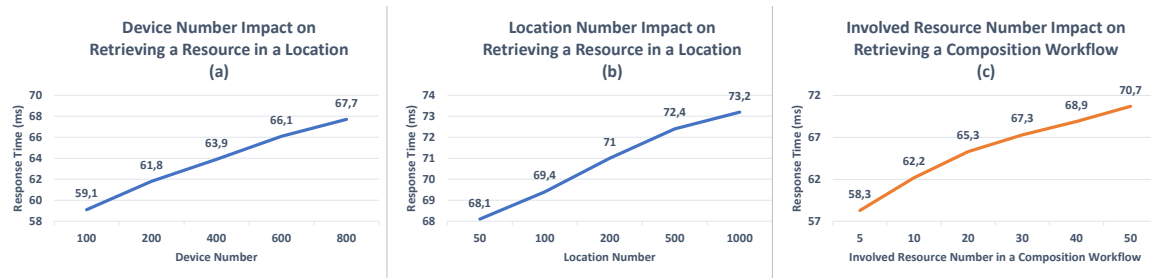


Fig. 4: Device and Location impact vs Involved Resources in a Composition Workflow impact

actuators, samplers as well as their observations, actuation, and sampling activities). We implemented WoR, and evaluated its effectiveness and performance in different resource graphs configurations, which have shown promising results. As future work, we seek to continue the evaluation of the ontology to test its clarity, to verify whether the names or labels used to describe the concepts/properties are not ambiguous for users (experts and non-experts), and its consistency, to check if the added concepts/properties generate inconsistencies within the ontology structure. Finally, we want to use the proposed solution in real Web-based environments as the Web platform OpenCEMS, and in ongoing Web-based projects provided by Open Group.

REFERENCES

- [1] Y. Perwej, K. Haq, F. Parwej, M. Mumdouh, and M. Hassan, "The internet of things (iot) and its application domains," *International Journal of Computer Applications*, vol. 975, no. 8887, p. 182, 2019.
- [2] K. Jacksi and S. M. Abass, "Development history of the world wide web," *Int. J. Sci. Technol. Res.*, vol. 8, no. 9, pp. 75–79, 2019.
- [3] S. B. Heilesen, "A short history of designing for communication on the web," in *Designing for Networked Communications: Strategies and Development*. IGI Global, 2007, pp. 118–136.
- [4] D. Guinard, V. Trifa, and E. Wilde, "A resource oriented architecture for the web of things," in *IOT*. IEEE, 2010, pp. 1–8.
- [5] J. Rao and X. Su, "A survey of automated web service composition methods," in *International Workshop on Semantic Web Services and Web Process Composition*. Springer, 2004, pp. 43–54.
- [6] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson, *Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more*. Prentice Hall PTR, 2005.
- [7] X. Feng, J. Shen, and Y. Fan, "Rest: An alternative to rpc for web services architecture," in *2009 First International Conference on Future Information Networks*. IEEE, 2009, pp. 7–10.
- [8] S. Mumbaikar, P. Padiya *et al.*, "Web services based on soap and rest principles," *IJSRP*, vol. 3, no. 5, pp. 1–4, 2013.
- [9] F. De Carvalho Diniz, "Composition of semantically enabled geospatial web services," Master's thesis, University of Twente, 2016.
- [10] A. Gyrard, S. K. Datta, and C. Bonnet, "A survey and analysis of ontology-based software tools for semantic interoperability in iot and wot landscapes," in *WF-IoT*. IEEE, 2018, pp. 86–91.
- [11] S. Mishra and S. Jain, "Ontologies as a semantic model in iot," *IJCA*, vol. 42, no. 3, pp. 233–243, 2020.
- [12] N. Seydoux, K. Drira, N. Hernandez, and T. Monteil, "Iot-o, a core-domain iot ontology to represent connected devices networks," in *EKAW*. Springer, 2016, pp. 561–576.
- [13] W. Wang, S. De, G. Cassar, and K. Moessner, "Knowledge representation in the internet of things: semantic modelling and its applications," *automatika*, vol. 54, no. 4, pp. 388–400, 2013.
- [14] A. Haller, K. Janowicz, S. J. Cox, M. Lefrançois, K. Taylor, D. Le Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, and C. Stadler, "The modular ssn ontology: A joint w3c and ogc standard specifying the semantics of sensors, observations, sampling, and actuation," *Semantic Web*, vol. 10, no. 1, pp. 9–32, 2019.
- [15] P. Gomes, E. Cavalcante, T. Rodrigues, T. Batista, F. C. Delicato, and P. F. Pires, "A federated discovery service for the internet of things," in *Proceedings of the 2nd Workshop on Middleware for Context-Aware Applications in the IoT*, 2015, pp. 25–30.
- [16] S. De, P. Barnaghi, M. Bauer, and S. Meissner, "Service modelling for the internet of things," in *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2011, pp. 949–955.
- [17] V. Charpenay and S. Käbisich, "On modeling the physical world as a collection of things: The w3c thing description ontology," in *European Semantic Web Conference*. Springer, 2020, pp. 599–615.
- [18] L. Daniele, F. den Hartog, and J. Roes, "Created in close interaction with the industry: the smart appliances reference (saref) ontology," in *FOMI*. Springer, 2015, pp. 100–112.
- [19] C. Wang, H. Ma, G. Chen, and S. Hartmann, "Evolutionary multitasking for semantic web service composition," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 2490–2497.
- [20] R. T. Fielding, R. N. Taylor, J. R. Erenkrantz, M. M. Gorlick, J. Whitehead, R. Khare, and P. Oreizy, "Reflections on the rest architectural style and" principled design of the modern web architecture"(impact paper award)," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, 2017, pp. 4–14.
- [21] J. Tihomirovs and J. Grabis, "Comparison of soap and rest based web services using software evaluation metrics," *Information Technology & Management Science (Sciendo)*, vol. 19, no. 1, 2016.
- [22] S. K. Datta and C. Bonnet, "Extending datatweet iot architecture for virtual iot devices," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2017, pp. 689–694.
- [23] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog *et al.*, "The ssn ontology of the w3c semantic sensor network incubator group," *Journal of Web Semantics*, vol. 17, pp. 25–32, 2012.
- [24] A. Haller, K. Janowicz, S. J. Cox, M. Lefrançois, K. Taylor, D. Le Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, and C. Stadler, "The sosa/ssn ontology: A joint w3c and ogc standard specifying the semantics of sensors, observations, actuation, and sampling," *Semantic Web-Interoperability, Usability, Applicability an IOS Press Journal*, vol. 56, pp. 1–19, 2019.
- [25] E. Mansour, R. Chbeir, and P. Arnould, "Hssn: an ontology for hybrid semantic sensor networks," in *Proceedings of the 23rd International Database Applications & Engineering Symposium*, 2019, pp. 1–10.
- [26] D. Bonino, F. Corno, and L. De Russis, "Poweront: An ontology-based approach for power consumption estimation in smart homes," in *International Internet of Things Summit*. Springer, 2014, pp. 3–8.
- [27] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, "Iot-lite: a lightweight semantic model for the internet of things and its use with dynamic semantics," *Personal and Ubiquitous Computing*, vol. 21, no. 3, pp. 475–487, 2017.
- [28] M. Lanthaler and C. Gütl, "Hydra: A vocabulary for hypermedia-driven web apis," in *LDOW*, 2013.
- [29] U. Arul and S. Prakash, "Toward automatic web service composition based on multilevel workflow orchestration and semantic web service discovery," *ijbis*, vol. 34, no. 1, pp. 128–156, 2020.