IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 26, 2018, accepted October 18, 2018, date of publication October 25, 2018,
date of current version November 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2878100

4W1H in IoT Semantics

GARVITA BAJAJ[1], RACHIT AGARWAL[2], PUSHPENDRA SINGH[1], NIKOLAOS GEORGANTAS[2], AND VALÉRIE ISSARNY[2]

[1]Grit Lab, IIIT Delhi, New Delhi 110020, India
[2]Inria, 75012 Paris, France

Corresponding author: Garvita Bajaj (garvitab@iiitd.ac.in)

This work was supported by the European Union's Horizon 2020 Programme, through the European Project, Federated Interoperable Semantic IoT/Cloud Testbeds and Applications (FIESTA-IoT), under Grant Agreement CNECT-ICT-643943.

ABSTRACT IoT systems are now being deployed worldwide to sense phenomena of interest. The existing IoT systems are often independent, which limits the use of sensor data to only one application. Semantic solutions have been proposed to support the reuse of sensor data across IoT systems and applications. This allows the integration of IoT systems for increased productivity by solving challenges associated with their interoperability and heterogeneity. Several ontologies have been proposed to handle different aspects of sensor data collection in IoT systems, ranging from sensor discovery to applying reasoning on collected sensor data for drawing inferences. In this paper, we study and categorize the existing ontologies based on the fundamental ontological *concepts* (e.g., sensors, context, location, and so on) required for annotating different aspects of data collection and data access in an IoT application. We identify these fundamental concepts by answering the 4Ws (What, When, Who, Where) and 1H (How) identified using the 4W1H methodology.

INDEX TERMS 4W1H, semantics, IoT, ontology.

I. INTRODUCTION

The number of IoT devices in the world is expected to increase to 30 billion by the end of 2025 with the rapid adoption of IoT technology in several domains including health, transportation, and manufacturing.[1] Several standalone IoT applications have been developed for these domains which rely on sensor data collected from dedicated '*things*'. The number of connected *things* is expected to reach 125 billion by the end of 2030, and they will generate a huge amount of sensor data.[2] This sensor data is multi-modal in nature comprising of various formats including video streams, images, and strings, and is often available via application platforms such as ThingsBoard[3] or SmartSantander.[4] The sensor data from multiple platforms, when combined, is used to actuate devices within applications and reused to extract human-interpretable knowledge.

Discovering these *things*, integrating and accessing their multi-modal, heterogeneous sensor data from multiple application platforms is a key factor towards building *smart* future applications [1]. Towards this, semantic solutions - ontologies have been proposed. Ontologies are defined as a "*well-founded mechanism for the representation and exchange of structured information*" [2]. As highlighted by Barnaghi *et al.* [3], ontologies are used to solve the issues related to data interoperability (multiple stakeholders accessing the sensor data unambiguously), data integration (combining one sensor's data with other sensor's data), data abstraction and access (generating knowledge for readability), search and discovery (locating physical devices and services), and reasoning and interpretation. These issues can be addressed at different levels of the Semantic Web Stack for IoT [4] - Modeling, Data Processing, and Services and Applications. Ontologies defined at the modeling level of the stack define concepts associated to *things*' characteristics and capabilities. These ontologies enable sensor search, sensor discovery, and data integration functionalities. The ontologies defined for the next level - Data Processing - propose concepts to enable data abstraction and access, and support description logics to reason, infer, and interpret the data. Ontologies defined for the last level - Services and Applications - enable service search, service discovery, and service composition. Ontologies may also define concepts catering to multiple levels of the stack. Ontologies are now commonly used to solve the issues associated with semantic interoperability in IoT domain because of the following benefits [5]: (1) exchanging

[1] https://goo.gl/F9zBp9
[2] https://goo.gl/NC5AUs
[3] https://thingsboard.io
[4] http://www.smartsantander.eu/index.php

data among systems, (2) providing interoperability among systems, (3) designing knowledge, (4) sharing knowledge, and (5) simplifying operations.

The existing ontologies defined for IoT are either too generic (causing loss of information because of coarse granularity), or limited to only a specific domain (verbose and hard to use) [6]. As technology progresses, additional *things*/information will be incorporated within IoT applications which will require more concepts to be introduced within ontologies (existing or new). This results in a need to introduce more concepts to make the ontologies more explicit with respect to applications. Thus, *ontologies need to be redesigned and formalized over time*.

Existing works have proposed the use of *unified* ontologies to tackle issues of interoperability and automation associated with heterogeneity of sensor data [7]–[9]. Such unified ontologies will still need to be redesigned and formalized with time. Also, multiple possible unifications developed by domain experts pose several challenges such as dealing with constantly redefining ontologies and a lack of standards [10]. Figure 1 illustrates an example scenario where the use of several possible unifications of ontologies causes issues. Consider, for example, two IoT testbeds deployed at different (or same) geographic locations and using different unified ontologies for building applications for smart-health and smart-building domains respectively. Accessing data from both the testbeds at a remote location (e.g., a remote server using data from both the platforms for developing another application) leads to the following challenges.

First, accessing data from a particular sensor based on context (such as location or time) requires discovering the sensor. Since there are two testbeds, they may have different hardware and software specifications to discover *things* and access data, and thus, different vocabularies. This problem is further exaggerated when multiple testbeds with multiple hardware/software specifications are involved. Unifying multiple ontologies from different testbeds in this case requires defining a common vocabulary to meet these specifications. Thus, there is a need to have a common discovery mechanism to access a particular sensor within a testbed based on its context to access its data.

Second, heterogeneous ontologies also cause issues in the integration of multi-modal sensor data from multiple testbeds. This results in complexity and variability in the information exchange. For example, Cloud 1 may store the temperature values as instances of the class `Temp` (defined by ontology A); while Cloud 2 may store them as instances of class `Temperature` (defined by ontology B). Moreover, the temperature values stored in the former case would correspond to body temperature values (smart-health domain) while the latter would represent room temperature values (smart-building domain). This difference in nomenclatures and heterogeneity of data leads to usability issues in developing cross-domain applications as developers would require prior knowledge of every data source and its associated ontology before selecting one for use.

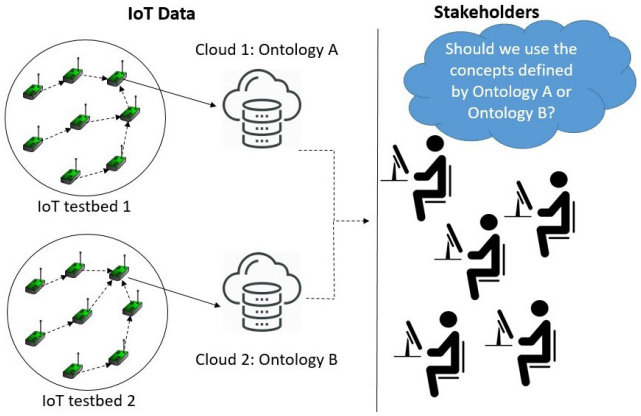

FIGURE 1. Challenges may arise with the use of semantic approaches in IoT applications. The lack of a common semantics may lead to interoperability and design issues across IoT applications and stakeholders.

Third, heterogeneity also causes problems in the design, interaction, and integration of automated solutions based on sensor data [11]. For example, with the growing complexity of building automation systems and available devices, ontologies are now being used to automate the registration process [12]. With multiple systems utilizing different ontologies, the automated registration process would also be affected. Therefore, it is important to develop a comprehensive ontology to specify hardware and software components to support automated approaches and avoid heterogeneity issues.

However, defining a comprehensive unified ontology for the IoT domain is equally challenging because of the following reasons:

- There are more than 200 domain-specific ontologies available [13]. Thus, it becomes difficult, even impossible, to come up with an exhaustive list of concepts for IoT systems as the field is constantly evolving with the introduction of new application areas.
- The unified ontology defined using existing ontologies will again consist of its self-defined (or borrowed) classes which would regenerate the problems listed earlier.

Although there have been some works on unification of ontologies for the IoT domain [14], [15], the problem of heterogeneity still persists as each of these unifications proposes their own terminology. Even though they follow certain best practices listed in the literature [5] to ensure that they are easily accessible for reuse, there are chances of conflicts in the concepts defined. We, therefore, shift our focus from defining a comprehensive/unified ontology to identifying the requirements of an extensible ontology for the IoT domain. Instead of proposing a new ontology, we focus on identifying the core concepts that are peculiar to every IoT testbed and application. Consider, for example, the two testbeds shown in Figure 1. Both these testbeds require concepts to answer the basic queries posed by a stakeholder including discovering a sensor based on its capabilities and accessing the sensor

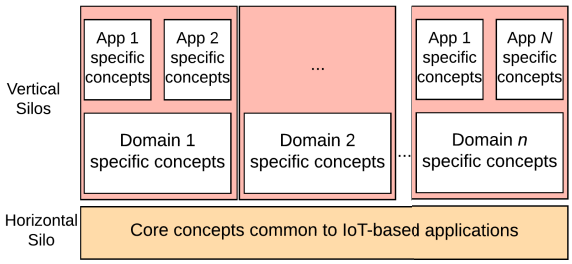

FIGURE 2. Distributing the concepts requirement of an IoT-based application into horizontal and vertical silos.

data, among others. These concepts constitute the horizontal silo of the semantic description for an IoT system while the domain specific and application specific concepts constitute the vertical silos of an IoT ontology as shown in Figure 2. In this study, we focus on the concepts required in the horizontal silo of IoT domain. For defining an application specific ontology in the IoT domain, the concepts in the horizontal silo can be extended with domain and application specific concepts while offering interoperable and heterogeneous data. However, application-specific ontologies are not the focus of this work and we only consider domains that define concepts associated with IoT related areas (details in Section II).

The two-fold contribution of this work is as follows: (i) We identify the fundamental concepts required for a horizontal IoT ontology by identifying the **competency questions** - "*what are the queries that experts will submit to a knowledge base to find answers?*" [16]. These competency questions are identified using the 4W1H methodology prevalent in the literature [17], [18]. By answering the 4Ws (What, Where, When, Who) and 1H (How) associated with an IoT application, we identify the basic concepts required to define an IoT ontology. (ii) Based on the concepts identified using the 4W1H methodology, we study the existing ontologies centered around these concepts. We classify these ontologies based on the competency question answered by them. Our proposed categorization and structure can be used by ontology developers to identify the concepts required for defining an extensible, core ontology for IoT.

ROADMAP

The rest of this work is organized as follows: We position our work in the literature and differentiate it from the existing IoT related surveys in Section II. In Section III, we briefly explain the functioning of an IoT application. This is followed by using the 4W1H methodology to identify the competency questions that may arise in an IoT application use-case. Based on these competency questions and their answers, we identify a structure within the proposed ontologies and study them in Section IV. We conclude our work with a brief discussion in Section V.

II. SCOPE OF THIS WORK

The field of IoT is rapidly evolving and is often confused with related areas and technologies such as Ubiquitous computing, Pervasive computing, Ambient Intelligence, WSN (Wireless Sensor Networks), M2M (Machine-2-Machine) communication, CPS (Cyber-Physical Systems), WoT (Web of Things), Cloud Computing, Big Data, and Context-Awareness. In this section, we highlight the differences, rather relationships, between these areas and technologies with IoT. We then proceed to discuss some of the existing surveys proposed in IoT and differentiate them from our work to highlight our contribution.

A. RELATED AREAS

The fields of Ubiquitous Computing, Pervasive Computing, and Ambient Intelligence were introduced before IoT and these technologies form the basis of IoT. Ubiquitous computing is a paradigm that refers to extending computing platforms to beyond just computers to provide anytime, anywhere, any device computing capability. This is done by utilizing pervasive computing which aims to adapt computing models to the environment by taking device contexts into consideration. In other words, pervasive computing "treats context as a first-class citizen and adapts computing models based on context" [19]. Ambient intelligence refers to technologies that allow devices to assist users in daily life activities by utilizing pervasive and ubiquitous computing paradigms. An example of ambient intelligence is the smart-home technology where sensing and computing devices interact and communicate with each other to actuate the environment to assist users. We refer the readers to [19] for a more detailed description of these technologies.

The correlations between WSNs, M2M, CPS, and IoT are discussed in detail by Chen *et al.* [20]. WSN and M2M are supplementary technologies that form the basis of IoT domain. In WSN, a network of spatially distributed autonomous sensing nodes with wireless communication capabilities is deployed to "monitor physical or environmental conditions, and to cooperatively pass their data through the network to a main location" [20]. These nodes are limited to performing sensing only. M2M refers to technologies that allow wireless/wired systems to communicate with other machines with similar capabilities. M2M technologies are limited to machines and currently require no human involvement. Building upon M2M, CPS enables strong coordination between computational (the 'Cyber' element) and physical elements of a bigger WSN system which can be used to develop interactive applications. IoT entwines these three technologies (WSN, M2M, and CPS) together by deploying nodes that are capable of performing sensing and computation; and also have actuation capabilities which makes it a smarter WSN network that relies on M2M and is capable of producing CPS. An extension of IoT to standard web technologies and protocols results in WoT. WoT uses standard Web technologies for the network of things - such as the use of `HTTP` to access things and locate them uniquely using URLs, REST protocol for communication, and JSON data representation format.

The goal of IoT is to provide anytime, anywhere and any-device services. Thus, cloud-based architectures have been

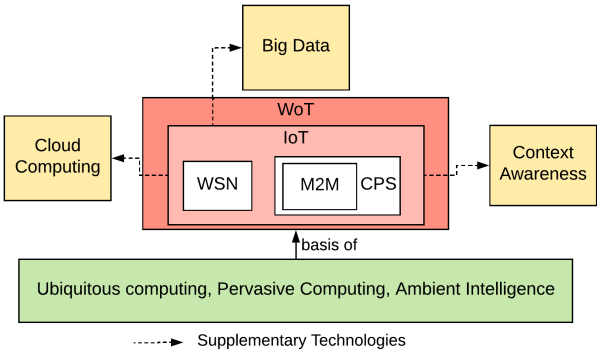

FIGURE 3. Relationship between IoT and related technologies.

proposed in the literature [21]. These cloud-based architectures rely on cloud computing technologies which provide criterion for service provisions with reduced costs and improved availability and fault tolerance. The cloud technologies are supplementary to IoT technology and support easy storage, management, processing, and analysis of large-scale sensor data. The large-scale sensor data collected by IoT systems usually comprise the following characteristics as highlighted by Ma *et al.* [22] - heterogeneity, inaccuracy, and data collection in real-time. Traditional data processing techniques, therefore, may not work on this data which is why IoT relies on big data technologies for data processing and analysis. Intelligence to IoT networks with the processed information can be incorporated using context-aware technologies that help in filtering, searching, and interpreting large-scale sensor data. Integration of context-aware technologies with IoT results in context-aware IoT. Figure 3 shows the relationship between the discussed areas and IoT.

B. OUR WORK

Several surveys on IoT and IoT-related technologies have been conducted in the past. For example, a survey on big data technologies and their impact on IoT is provided in [23]. Petrolo *et al.* [24] review context-aware technology and supported IoT applications. A comprehensive review of surveys until 2016 in IoT techniques was presented by Gil *et al.* [25] to provide a global, comprehensive view of how techniques belonging to multiple layers of IoT architecture can be integrated to provide services to final users. However, these existing works focus on technologies and their use with IoT but fail to identify the challenges associated with data heterogeneity within IoT systems. Although the survey by Karkouch *et al.* [26] talks about data quality in IoT systems, it fails to highlight the interoperability and heterogeneity of data. It only discusses the factors that affect data quality but does not consider the semantic aspect of it. The semantic aspect is superficially addressed in the survey by Szilagyi and Wira [4] that ''covers certain aspects of semantic technologies (ontologies) used in IoT but does not provide an in-depth analysis of ontologies and schemas''. In this work, we present a detailed analysis of the ontologies proposed for IoT and IoT related domains to capture the key concepts that can be used to solve the heterogeneity issue within IoT systems. This is done by first identifying the competency questions as discussed in the next section.

III. IDENTIFYING THE COMPETENCY QUESTIONS

In order to identify the competency questions, i.e., the queries that stakeholders might submit to an IoT knowledge base (data repository), we must first understand how an IoT application works. In this section, we briefly discuss the working of an IoT application and then identify the competency questions.

A. HOW IOT APPLICATIONS WORK?

Gubbi *et al.* [27] define IoT as an ''interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications''. The applications developed using IoT comprise three components: ***(i) Hardware***: the *things* (''made up of sensors, actuators, and embedded communication hardware'') that sense the environment and act upon it; ***(ii) Middleware***: for collecting, storing and analyzing the collected sensor data, and ***(iii) Presentation***: tools to understand, visualize, and interpret the data on different platforms which can be designed for different applications. For the rest of this paper, we refer to an application based on IoT technology as an IoT application.

An IoT application facilitates interacting with the hardware components using the presentation tools via the middleware. The hardware or the '*things*', attached to a platform, are either static or moving and collect data across a geographical area of interest. The *things* may be connected directly or via a gateway to a middleware which receives/sends data from/to *things*. The data collected at the middleware is then used to generate high-level abstractions (knowledge) and are further used for actuations or generating wisdom for other applications [28]. The knowledge generated from *things* is application-specific and thus, the concepts required to generate this knowledge are also application-specific. For example, for a smart-health application, knowledge would comprise of health-related information such as health diagnostics or fitness information, while for smart-transportation application, knowledge could include inferring average vehicle speed and finding carbon emission rate. This knowledge is generated implicitly (using libraries or frameworks residing within the middleware) or explicitly (using third-party middlewares outside the scope of IoT application middleware) [28].

In this work, we limit our discussion to identify concepts required to answer competency questions associated with data collection and data access (raw sensor data and inferred knowledge) from the perspective of the three IoT components - hardware, middleware, and presentation. We only focus on ontologies that define the core concepts - the concepts required to answer queries posed by experimenters/stakeholders to fetch IoT data from the IoT

application middleware. These aspects comprise the horizontal silo of an IoT-based application. We do not focus on concepts required to generate the knowledge as the process of inferring knowledge is application dependent.

B. COMPETENCY QUESTIONS: 4W1H METHODOLOGY

4W1H [17], [18] is a popular approach that is used to describe the basics of an event/situation. To define the comprehensive list of concepts for an IoT ontology, we answer the five basic questions - four **W**s (**W**hat, **W**here, **W**hen, **W**ho) and one **H** (**H**ow). These five questions comprise the exhaustive list of *competency questions* - queries submitted by experts to a knowledge base to find answers [16]. The answers to these questions are required to identify the core IoT concepts. We now present these variables and the corresponding competency questions relevant to the IoT domain in detail below:

- **What:** *What* are the *things* involved in IoT applications? Are the *things* actual physical devices or virtual entities? Are these *things* used for sensing or actuation? What are the capabilities of these *things*? What is the lifetime of these *things*? What is the phenomenon of interest measured by these *things*? What are the conditions under which these *things* must collect data? Should the data be collected under certain circumstances only? The answer to 'what' of an IoT application requires concepts to identify the *data sources* and their *context* (e.g., the mobility of the sensor, the activity being performed, whether measurements were taken by devices automatically or was there some human intervention). This source is the *sensor* or the *augmented entity* which is embedded in a *platform* that can be a part of a *testbed*. Thus, to answer the 'what' question, an IoT ontology must include concepts for sensors, their context, and related entities (such as platform and testbed).
- **Where:** *Where* are the *things* located? Is the location a specific geo-coordinate, a spatial region, or a point of interest (a landmark)? Is the location absolute or relative (w.r.t. a testbed)? How should the stakeholders refer to the location of a *thing* to access its data? To answer these questions, the IoT ontology must incorporate different ways to locate *things* to allow stakeholders to access location-based data. There must be concepts to define and incorporate *location* of data sources.
- **When:** *When* are the *things* collecting data? Are they collecting data at specified *timestamps*, at a regular frequency, over a span of time? Is data collection event-driven? What is the granularity of time duration? Is there a concept of a timespan? Is time specified using a universal format (UTC) or in local time format? An IoT ontology should provide concepts to support different formats for defining the *time* of data collection.
- **Who**: *Who* are the users/applications that are allowed to access information from IoT systems? IoT technology is still evolving with multiple options available for access protocols and data exchange. Allowing interoperability between protocols and data exchange mechanisms may introduce vulnerabilities in the system. Therefore, it is important to authenticate users (and applications alike) before granting them access to sensitive data collected by IoT systems. To ensure this, an IoT ontology must include concepts to categorize different *users* and their *access levels*.
- **How:** *How* do users/applications interact with *things*? How are new *things* deployed within an IoT application? How are *things* safely exposed to authenticated users and applications once the required data is collected? How are commands sent to things for actuations? There should be concepts to support consistent *services* for providing two-way interaction between users/applications and *things*. Two-way interaction is required to allow users/applications to send commands to actuators for controlling the surroundings (interaction from user/application to things) and to securely access the data from sensors (interaction from things to user/application). Ensuring secure and standard access is important to handle several vulnerabilities that are introduced in IoT systems because of the constrained interactions and multiple protocols available. Providing standardization also helps in automating the data collection and data access process. Thus, it becomes important to define and introduce concepts for enabling services. Services provide consistent functionalities in the form of standardized interfaces or to interact with devices in an IoT system.

We now answer these questions by discussing the recent ontologies and comparing their effectiveness in supporting an IoT system with respect to how well they can answer the competency questions.

IV. ONTOLOGIES AVAILABLE

In this section, we study the ontologies that include concepts required to answer the competency questions discussed above. We did a keyword-based search on several scientific research-referencing websites to identify the state-of-the-art ontologies. Various keywords and their combinations were used for search, such as ''ontology'', ''semantic'', ''modeling'', ''iot'', ''access'', and ''temporal''. The search results included ontologies prevalent across several domains (such as Wireless Sensor Networks [15] and Manufacturing Processes [29]) and application areas (such as Building Management Systems (BMS) [30], indoor navigation [31], and smart-homes [32]). We filtered these results by considering only recent (proposed since 2012) and generic ontologies that encompass concepts from a broader perspective and are not restricted to a certain domain/proprietary application.

We have categorized these filtered ontologies based on the concepts introduced by them. 6 main categories (Sensor, Context, Time, Location, Access Control, and Service) are identified to answer the 5 competency questions - *W*hat, *W*hen, *W*here, *W*ho, and *H*ow. Some papers have been included

TABLE 1. A brief overview of sensor based concepts covered by different sensor ontologies proposed since 2012. A ✓represents that concepts are available and a ✗ indicates the absence of the concepts.

Ontology	Year	Parent Ontologies	Concepts Included										Supports			
			Sensor	Sensor observation	Observation Characteristics	Stimulus	Actuator	Platform	Testbed	Procedure	Sensor Capabilities	Service	Sensor Discovery	Sensor Data Management	Sensor Description	Sensor Registration
SCO [37]	2012	SUMO, DUL, *.	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✗	✓	✗
Nambi et al. [9]	2014	SSN$^+$, GeoNames, OWL-S.	✓	✓	✓	✗	✓	✗	✓	✗	✓	✓	✓	✓	✓	✗
Xue et al. [38]	2015	*-NA-*	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗
MyOntoSens [15]	2015	SSN$^+$, OntoSensor, WSSN, SensorML, QUDT.	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓
M3 [8]	2015	SSN$^+$, WGS84, Domain Specific ontologies (FOAF, DogOnt, Cooking, etc.).	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
Hirmer et al. [39]	2016	SensorML.	✓	✓	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓
ContQuest [40]	2016	*-NA-*	✓	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	✓	✓	✗
IoT-Lite [41]	2016	SSN$^+$, QU, WGS84.	✓	✓	✗	✗	✓	✗	✓	✗	✗	✓	✓	✗	✓	✗
SSN [42]	2017	QUDT, O&M, OWL-Time, DUL, GeoSPARQL.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗
SmartOntoSensor [43]	2017	SSN$^+$, SensorML, CoDAMoS, OWL-Time.	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓	✗	✗	✓	✗

+: First version of SSN (proposed in 2005).
* All parent ontologies used are not explicitly mentioned in the paper. The ontology is also not publicly available for verification.

in more than one category if they cover multiple concepts. We now present our study of existing ontologies to answer the different competency questions based on these categories. For each ontology studied, we summarize its major concepts and parent ontologies (if any - from which they inherit concepts[5]) in the form of a table. The studied ontologies can further be evaluated on the basis of multiple parameters [33] such as accuracy, clarity, and computational efficiency; however, comparing these ontologies is not the focus of this work.

A. ANSWERING "WHAT"

Addressing the "what" aspect in IoT requires discovering and identifying *things* based on their capabilities and *contexts*. *Things* in IoT refer to devices, agents, or softwares that connect and exchange data with the outside world. These *things* monitor/control contexts. Monitoring contexts requires answering the "what" aspect of IoT application - what is the sensor reading under some condition? Controlling contexts, however, requires actuation and is covered in the "how" aspect (covered in detail in Section IV-E). Thus, to answer the competency questions associated with "what" aspects of an IoT application, we study the concepts that define sensors and contexts. We divide this section into subparts to address each of these concepts separately.

1) SENSOR ONTOLOGIES

Several ontologies have been proposed in the literature for sensor technologies. They aim to solve the problem of heterogeneity associated with hardware, software, and data management aspects of sensors. Several ontologies have been proposed for different domains that utilize sensor data such as **SAREF** for smart appliances [34], **Brick** for smart buildings [30], and ontology by Dey and Dasgupta [35] for smart energy meter. For this study, we have only considered the ontologies proposed since 2012[6] that include generic concepts associated with different aspects of sensor data as highlighted in Table 1. The table summarizes the different concepts included in these ontologies and identifies the different problems they tackle. We now present a detailed explanation of the different concepts included in these ontologies and the functionalities they support.

As shown in Table 1, all sensor ontologies include concepts to define 'Sensors' and 'Sensor Observations' where a 'sensor' refers to a physical or virtual resource that is used to sense a phenomenon. A physical sensor refers to an actual physical device that is capable of sensing an environmental phenomenon while a virtual sensor refers to a virtual device (also referred to as a composite device) that combines raw data from multiple physical sensors to sense bigger/complex phenomena. An example of a virtual sensor includes a weather sensor that uses data from temperature and humidity sensors

[5]For detailed description of the respective parent ontologies, we refer the readers to the corresponding inheriting ontologies discussed.

[6]For a review of the sensor ontologies proposed before 2012, we refer the readers to the study by Eastman *et al.* [36]

to infer weather conditions. The data/readings captured by these physical and virtual sensors are the 'Sensor Observations' which may comprise a single value or multiple values (a tuple of sensor readings) that may belong to different data types (such as numeric and boolean). The sensors may also detect 'stimulus' which is a real-world event that triggers a sensor. An example of a stimulus is a reading from a smoke detector that triggers a temperature sensor to check if there is a fire.

These sensing devices used for data collection also vary in terms of their 'sensor capabilities'. The sensor capabilities are mainly described using metadata information (such as their range, the accuracy of sensor data collection, and the manufacturer). Defining these sensor capabilities can help experts in querying a subset of sensors within a 'testbed' (an abstract collection of *things*) comprising the IoT sensor network. The sensors associated to testbeds may be deployed on a 'platform' which serves as a host for a sensing device and is helpful in identifying sensors. Consider, for example, a temperature sensor deployed on a tree (acting as a platform) within a large forest area (serving as the testbed). In some cases, to further filter out sensors based on the characteristics of sensor observations made, an IoT ontology can define 'observation characteristics', such as measurement units, accuracy or latency of sensor observations, to help experts narrow down their queries.

To assist with remote management and registration of *things*, some ontologies include concepts to support automatic 'sensor registration' where new *things* (yet to be associated with a platform) send additional semantic data (sometimes inferred with reasoning capabilities) to the testbed server/cloud to identify and register themselves along with their capabilities. To help experts easily search and query the registered *things*, ontologies support concepts to define 'services' which provide a medium to interact with *things*. These services (described in detail in Section IV-E) are sometimes used to facilitate message passing from experts to 'actuators' present in an IoT system so that they can perform some action based on defined 'procedures'. A 'procedure' refers to a computational method that uses sensor observations to generate actuations. As an example, consider a smart-home testbed that consists of a temperature sensor and an air conditioner. An expert can use a service to query temperature readings from a particular sensor, and define a procedure to actuate the air conditioner if the observations captured are below a certain threshold value. The different concepts defined in these ontologies support different functionalities such as sensor discovery and sensor description. Some ontologies also include concepts to support sensor data management (in the form of efficient storage) while some provide dynamic support for registering new sensors with IoT systems. We now discuss the identified ontologies in detail below.

In 2012, W3C (World Wide Web Consortium) proposed a standard ontology - **SSN** (Semantic Sensor Network) [42] for describing sensors and sensor observations. The ontology has been modified recently (October 2017) and now includes a comprehensive list of concepts to describe sensor capabilities such as accuracy, range, and resolution. This facilitates sensor discovery based on sensor capabilities. Further, the new version also uses concepts from `GeoSPARQL`[7] ontology to define location-related concepts to facilitate sensor discovery as per spatial requirements.

The concepts proposed by Xue *et al.* [38] to define sensor capabilities differ from the sensor capabilities defined in SSN. They classify sensor capabilities as `static` and `dynamic`. They also categorize sensors as `normal` or `advanced`. They introduce concepts to deal with issues of sensor management and data sharing in sensor networks. However, their ontology is limited to defining concepts for indoor locations only (building rooms and floors) and supports only a small number of sensor types in terms of the phenomena they sense. This provides limited semantic support for only a small number of sensor features. **M3** ontology, proposed by Gyrard *et al.* [8], [44], overcomes this limitation by providing support for several sensors, observation values (multiple data types), their units, and domains. Using rules defined on this data, M3 supports reasoning on sensor data to infer contextual information. M3 ontology was proposed as an extension to W3C's initial SSN ontology.

Since the platform to which sensors are attached can be mobile, issues arising with dynamicity and sensor discovery were initially dealt by the OntoSensor [45] ontology proposed in 2005. The ontology extends concepts from SensorML,[8] ISO-19115,[9] and SUMO [46], to provide concepts for the identification of sensor categories, behavior, relationships, functionalities, and meta-data regarding sensor characteristics, performance, and reliability. OntoSensor aims to support interoperability and ontology-based inferences that require aspects of physical sensing to be incorporated in the ontology definition. The ontology is heavy-weight, has high usage complexity, does not include concepts to define services, and lacks the ability to extensively describe sensor observations. These limitations were solved by **MyOntoSens** [15] which provides a generic and exhaustive ontology for describing sensor observations and capabilities to reason over collected data. MyOntoSens has been proposed for the domain of wireless sensor networks (WSN) and borrows several concepts and relationships from existing ontologies including OntoSensor [45], SSN [42], and QUDT [47] which makes it applicable to the IoT domain also. MyOntoSens provides concepts to support sensor discovery and sensor registration with IoT systems.

Hirmer *et al.* [39] propose an ontology to support dynamic registration and bindings of new sensors to platforms. They borrow concepts of sensors, their characteristics, associated properties from SensorML, and introduce an additional concept of 'adapter' associated with every sensor. The

[7] http://www.opengeospatial.org/standards/geosparql
[8] https://en.wikipedia.org/wiki/SensorML
[9] http://bit.ly/2nedvM1

borrowed concepts support sensor discovery while the newly introduced concepts provide/compute additional information about sensor data. For example, adapters are used to compute the average quality of sensor data values from the quality of the sensor provided by the manufacturer and the staleness of values generated by the sensor. This additional information is then used to build a repository of different possible sensor bindings - bindings of different sensors with different adapters - which is then used to automate registration and binding tasks.

Shi *et al.* [37] identified the problems associated with inconsistencies in concept definitions among existing ontologies and proposed a framework to overcome them. The ontology associated with their framework, **Sensor Core Ontology** (SCO), borrows concepts from existing sensor ontologies and allows addition of new concepts, thereby supporting extensibility. The ontology focuses mainly on concepts related to sensor data where each sensor observation is associated with time, space (location), and a theme (phenomenon being sensed). Similar to this, Bermudez-Edo *et al.* [41] suggest that for data analytics in an IoT application scenario, only some concepts are required from the SSN ontology to query the datastore. According to them, it is inefficient to borrow all concepts as it increases query time. Hence, they propose a lightweight ontology by borrowing concepts from existing ontologies (such as SSN and WGS84) to define the basic concepts required for accessing the sensor data.

Every generic ontology listed above aims to solve some aspects of sensor data collection from its limited perspective. An individual ontology is still unable to address all aspects associated with sensor data collection and thus, developers must choose an ontology (or a combination of them) based on their application requirements. This highlights the problems associated with heterogeneous ontologies listed earlier.

2) CONTEXT ONTOLOGIES

Most of the IoT applications aim to sense environmental phenomena and take suitable actions to actuate the environment, or infer high-level knowledge (contexts) about the environment. The knowledge and the contexts inferred by one IoT application can be used for other applications as well. Thus, to enable knowledge sharing, interoperability, and extensibility in an IoT application, it becomes important to define concepts associated with contexts. As defined by Chen *et al.* [48], contexts are "*used to describe places, agents, and events*". Contexts can be classified as external or internal; and physical or logical [49]. In this work, we study the context ontologies proposed for context-aware systems that can be extended to the IoT domain also. The purpose of these context-aware ontologies is to effectively label contextual information collected from sensor devices in the form of sensor data. This contextual information can also be used to identify services relevant to a *thing*/actuator/user at a given location and time.

Context ontologies can be categorized on the basis of several aspects [28]: (i) the type of context monitored - dynamic or static; (ii) the type of information available - primary or secondary; and (iii) the level of context awareness offered - personalized, active, or passive. Dynamic contexts include information that varies quickly with time (such as location of a mobile sensor or the activity of sensor platform) while static contexts refer to information that does not vary with time (such as a user's profile information). Primary contexts refer to information that is extracted from the raw data available while secondary contexts are inferred using a combination of raw data and/or primary contexts. In terms of the levels of context awareness offered, personalized context awareness refers to the context information that is entered manually by a user (e.g., the location at which to trigger a certain action). Active context awareness is offered by systems that continuously sense the environment and autonomously take actions (e.g., smoke detectors monitoring continuously in a room and automatically informing a fire station) while passive context awareness is offered by systems that sense the environment automatically but provide a list of possible actions to be taken instead of deciding an appropriate action themselves (e.g., monitoring a refrigerator door and alerting the user in case it has been open for a long time). Depending on the requirements of an IoT application, different context ontologies are used. A list of context ontologies studied in this work is shown in Table 2.[10]

In order to define context, four pieces of information have been defined as primary context - location, activity, time, and identity [56]. In this work, we study the existing context-aware ontologies across the different categories identified above and the different information considered as primary contexts. Table 2 presents a summary of our study. A detailed description of these ontologies is presented next.

It must be noted that domain specific context-aware ontologies are more prevalent as contexts inferred in context-aware applications (including IoT-based applications) are highly dependent on the domain in consideration. As an example, for labeling a user activity, we need a complete understanding of activities possible in the domain as activities performed on a university campus are very different from the activities performed in a smart-home environment. Thus, better concepts for semantic labeling of contexts can be identified with domain-specific ontologies but we only focus on generic ontologies for the horizontal silo of IoT applications.

In IoT applications, contexts are often important to correctly interpret sensor data [28]. Thus, as the first step towards semantic labeling of contexts in IoT applications, Baldauf *et al.* [49] propose the common architecture principles of context-aware systems based on the classification of contexts: external (physical) or internal (logical). External or physical contexts are those that are measured using physical sensors, while internal or logical contexts are those that are explicitly specified by users or captured by monitoring user interactions (e.g., a user's goal or emotional state).

[10]For a detailed study on context-aware ontologies in the domain of IoT before 2012, we refer the readers to [28].

TABLE 2. A summary of the existing context-aware ontologies in terms of primary contextual information offered and the level of context awareness provided. A ✓indicates that concepts are available and a ✗ indicates the absence of concepts.

Ontology	Year	Parent Ontologies	Primary Contextual Information					Type of context	Level of context awareness
			Location	Time	Identity		Activity		
					User	Thing			
iConAwa [50]	2012	-*NA*-	✓	✓	✓	✓	✗	Dynamic + Static	Active + Personalization
Rodriguez et al. [51]	2012	-*NA*-	✓	✗	✓	✗	✗	Static	Personalization
DCON [52]	2013	DUHO, Domain ontologies (such as DPO and NIE)	✓	✓	✓	✓	✓	Dynamic	Active + Passive
COAC [53]	2013	-*NA*-	✓	✓	✓	✓	✗	Dynamic + Static	Active
Nambi et al. [9]	2014	SSN^{+}, GeoNames, OWL-S.	✓	✓	✓	✗	✗	Dynamic	Passive
CACOnt [54]	2015	FOAF, OWL-S.	✓	✗	✓	✓	✓	Dynamic	Personalization+ Active
Gao et al. [55]	2016	SOUPA-Location, GCACO.	✓	✓	✓	✓	✗	Static	Personalization
SmartOntoSensor [43]	2017	SSN^{+}, SensorML, CoDAMoS, OWL-Time.	✓	✓	✓	✓	✓	Dynamic + Static	Passive

+: First version of SSN (proposed in 2005).

To further describe the primary concepts for contexts (locations, users/*things*, and events), Chen *et al.* [48] proposed a context-aware ontology **COBRA-ONT** in 2003 based on their COntext BRoker Architecture (COBRA) for smart-spaces (such as college campus). They describe places using `<lat, lon, string-name>` with different constraints. The 'agents' (such as *things* and services/procedures) are located in places and play certain roles to perform some activities. An important contribution by the authors is the broker architecture that is used to acquire and reason over contextual information from mobile devices in order to reduce the burden of developers.

Another system - **iConAwa** [50] also aims to provide context aware services to mobile devices in a multi-agent system. They propose a context ontology with concepts to define locations, *things* (mobile agents), and users that enables dynamic and personalized contextual awareness. It lays a strong foundation for mobile systems by proposing concepts to define a user's personal profile and mobile device characteristics such as brand, model, and CPU speed. To support contextual information across multiple domains, Rodríguez *et al.* [51] propose a multi-dimensional ontology which is capable of representing multiple dimensions, such as user contexts and application domains, with changing attributes such as user interests. This allows easy integration with other ontologies to solve multi-disciplinary problems, and allows identifying services relevant to users based on their interests. The relationships between user locations and contexts have also been modeled by **DCON ontology** [52] which uses sensor data to model two aspects of user's context - (i) live contextual information as inferred using sensor data (e.g.: temperature measured at a given region at a given time) and (ii) situation which describes a recurring live context (e.g.: rainy weather for two days). DCON ontology also identifies the environment around a user (peers and upcoming schedule) to identify services that match user requirements.

Expanding to the entire IoT domain, Nambi *et al.* [9] have proposed the use of a context ontology to provide ''context awareness and interoperability during service discovery and composition''. This allows easy addition and identification of IoT devices in a large testbed for enabling smart services to users. A more generic, hierarchical context-aware ontology - **COAC** - is proposed by Kayes *et al.* [53] to include core contextual concepts like location, time, and user in the upper (root) ontology, while allowing users to incorporate domain-specific ontologies in the lower layer of the ontology. This allows COAC to be extensible while targeting domain specific applications. **CACOnt** [54] is another ontology that aims to incorporate contextual information in a hierarchical manner to support generic context-based concepts (root level) and concepts specific to multiple domains (at the next level of hierarchy). This hierarchical grouping reduces the number of concepts required for an application, subsequently limiting the processing time and improving accuracy. Another similar hierarchical grouping of context-aware concepts is proposed by Gao and Dong [55]. Their proposed ontology is split into two levels - the first level consists of generic context-aware concepts such as location and time; and the second level consists of a personalization ontology that comprises of concepts to define user preferences. A combination of the concepts in these two levels allows their ontology to achieve user-level personalization for context-awareness. To account for the increasing sensing capabilities of mobile devices, and therefore, the large amount of contextual information that is generated by collected sensor data, Ali *et al.* [43] proposed a mid-level ontology **SmartOntoSensor**. SmartOntoSensor includes concepts to define contexts based on knowledge generated from sensor data while keeping it extensible to define new concepts as they evolve. As of now, SmartOntoSensor includes concepts to define the fundamental concepts required to identify a context - location, time, user/device, and activity.

However, as mentioned earlier, contextual information is largely domain-dependent. Consider, for example, the concept 'activity'. This concept can be used to define high-level physical activities (such as sitting, sleeping, or eating) of a patient in a smart-health application; while in a smart-education application, an activity can refer to finer activities (such as reading, writing, or playing). Incorporating the generic concepts in the horizontal silo of an IoT ontology help testbeds to solve the heterogeneity and interoperability issues in IoT data integration.

TABLE 3. Concepts and functionalities supported by different location ontologies proposed in the literature. A ✓indicates that concepts are present and a ✗ indicates the absence of concepts.

Ontology	Year	Parent Ontologies	Location Type	Region	Spatial Entity	Geo-coordinates	User	Device	Location characteristics	Accessibility Constraints
iConAwa [50]	2012	-*NA*-	Indoor + Outdoor#	✗	✓	✓	✓	✓	✓	✗
Brown et al. [57]	2013	-*NA*-	Indoor	✗	✓	✗	✓	✗	✓	✓
Nambi et al. [9]	2014	SSN+, GeoNames, OWL-S.	Indoor + Outdoor	✓	✓	✓	✗	✓	✓	✗
Kim et al. [58]	2015	-*NA*-	Indoor + Outdoor	✗	✗	✓	✓	✓	✓	✗
IoT-Lite [41]	2016	SSN+, QU, WGS84.	Outdoor*	✓	✗	✓	✗	✓	✗	✗
Gao et al. [55]	2016	SOUPA-Location, GCACO.	Indoor + Outdoor#	✓	✓%	✓	✗	✓	✗	✗
iLoc [31]	2016	QUDT, W3C Geo vocabulary, vCard.	Indoor	✗	✓	✗	✗	✗	✗	✓
SeCoMan [59]	2016	-*NA*-	Indoor	✓	✓	✓	✓	✓	✓	✓
ContQuest [40]	2016	-*NA*-	Indoor + Outdoor	✓	✗	✗	✗	✓	✓	✗
SmartOntoSensor [43]	2017	SSN+, SensorML, CoDAMoS, OWL-Time.	Indoor + Outdoor#	✓	✓	✓	✓	✓	✗	✗

+: First version of SSN (proposed in 2005).
#: The ontology does not explicitly mention that it provides concepts for indoor and outdoor locations but the concepts provided can be used to define both indoor and outdoor locations.
*: The ontology does not explicitly mention that it is limited to outdoor locations, but because of lack of concepts to define spatial entities, it is limited to defining geographical regions and points only which makes it difficult to describe indoor locations.
%: Supports limited spatial entities.

B. ANSWERING "WHERE"

The "where" aspect of answering questions related to data collection and data access is related to the location of *things*, platforms, and sensor data collection in IoT testbeds. Locations are used to describe the spatial context (partly, physical context) of users/devices. Although a subset of context, we consider location ontologies separately in detail as they have been used in several different domain areas beyond IoT technologies, such as defining cultural heritage [60], urban planning [61], and indoor navigation [57]. Semantic solutions for describing location information are also a key factor enabling Human Space Interaction (HSI) - which assist in providing context-aware services with IoT solutions [62].

The location ontologies (semantic solutions) are used to describe the geography of testbeds and locations of *things*/users in an IoT application. These locations and geographies include indoor locations (such as buildings or rooms) and/or outdoor locations (such as a park or a river). These locations are often described using geo-coordinates (a single point in space), as a region (using a polynomial shape defined as a set of coordinates), or as a spatial entity (using a well defined label for a landmark such as building, river, park, floor, and room). The location class is associated with metadata information that describes location characteristics such as its accuracy, name, nearby landmarks, and associated services. Each location/geography is associated either to a *thing* (device) or a user of an IoT application. These users and *things* may or may not have access to a certain location for security reasons or otherwise (e.g., a visitor in a hospital may not have access to patients' rooms). These accessibility constraints are defined within the location ontology using additional properties and concepts. Table 3 summarizes the different concepts provided by location ontologies studied in this work.

As mentioned earlier, location ontologies with generic concepts have been proposed in multiple domains. These ontologies are used across domains including IoT. The W3C Geospatial Incubator Group proposed the **WGS84** ontology [63] in 2003 as a basic location ontology to describe abstract concepts for 'SpatialThings' such as buildings, people, and 'TemporalThings' such as events or time durations. It also describes geographical locations of these spatial and temporal things with concepts for defining 'latitude', 'longitude', and 'altitude'. Concepts defined in WGS84 are inherited from abstract concepts for defining location sub-classes specific to a system. A more descriptive location ontology is provided by Flury *et al.* [64] for context-aware services as they identify location to be a common denominator for modeling services in context-aware environments. They provide a generic ontology for location concepts to define "device-based services encountered in ubiquitous computing environments". They provide abstract mathematical models to categorize the different location solutions (like Cartesian coordinates and numerical estimation techniques) used to define location information. The different models considered are as follows: (i) *Geometric models* comprising of Cartesian coordinates (geo-coordinates of a location/region); (ii) *Set-theoretic models* for defining location as an element of a set such as cellular location and WiFi AP location (spatial entities); (iii) *Graph-based models* for defining locations in physically grounded networks and social networks (regions) and (iv) *Semantic models* for defining locations defined using

human-friendly notations. Our work is limited to semantic models proposed in the literature that are used for context-aware environments.

iConAwa [50] system, as discussed earlier, is a multi-agent system for mobile devices aimed at providing context-aware services. The system proposes a location ontology that incorporates concepts for defining points of interests (geo-coordinates of a place) in an indoor setting and labeling them (e.g., a meeting room, a pantry). These POIs are often related to mobile agents and mobile users. The concepts are limited to spatial points and their labels (user-defined strings). There are no concepts to define spatial regions or entities which limits them to applications based on geographic points only. Another ontology defined for indoor settings (indoor navigation) was proposed by Brown *et al.* [57]. Their proposed ontology comprises of spatial entities split into space units (e.g., a building is split into multiple floors and a floor is split into multiple corridors) which are described using metadata information. The ontology also defines extensive concepts to support accessibility constraints from multiple aspects - users (who have access to a space unit), direction (which directions users can access while navigating indoors), or temporal (how many times can a space unit be accessed by a user). Similarly, the location concepts included in the **ContQuest** [40] ontology also allow a location to be located inside another location (e.g.: a room can be described as located inside a building, which can further be described as located inside a university campus). A major limitation of ContQuest is that it describes locations as regions only and therefore, serves a smaller set of applications.

To provide semantic solutions for both indoor and outdoor locations in IoT applications, Nambi *et al.* [9] proposed a location ontology which consists of concepts to define all forms of locations - coordinates, spatial entities, and regions. They associate these locations with IoT devices and also provide concepts to define some location characteristics (such as neighboring and nearby locations). To further incorporate more details about location characteristics, Kim and Kim [58] proposed an ontology that defines concepts to enable a reasoner to improve the accuracy of location prediction in indoor and outdoor settings. Since in most IoT-based applications, location information is often predicted from data collected using sensors (such as GPS and WiFi), Kim and Kim [58] rely on the use of sensor data collected from mobile devices of users and integrate it with location information from GPS to improve the accuracy of location prediction. They include concepts to define users and devices along with GPS logs and other sensor data that help in improving the location estimation. The lightweight sensor-based **IoT-Lite** [41] ontology and the ontology proposed by Gao and Dong [55] only comprise of concepts to define locations as a point or a region. This limitation also restricts them to a small set of applications.

The **iLoc** [31] ontology was also proposed for indoor locations (indoor building navigation). The ontology follows some of the best practices for defining a new ontology. It uses concepts borrowed from several existing ontologies - QUDT,[11] W3C Geo vocabulary, and vCard[12] and also supports extensibility. Since the root concept 'Location' is borrowed from W3C Geo vocabulary, iLoc can also be extended to provide navigation for outdoor locations. **SeCoMan** [59] ontology also provides concepts for both indoor and outdoor locations while providing support to define locations as a point, spatial entity, or a region. **SmartOntoSensor** [43] ontology for multi-agent systems also proposes location concepts to define locations of mobile devices.

C. ANSWERING "WHEN"

The ''when'' aspect is used to describe the temporal context of IoT sensor data collection and knowledge generation. The sensor data and inferred contexts in an IoT application evolve with time. This dynamicity in the collected and generated information is impossible to capture using binary relations, so N-ary relations are required for time-related concepts. E.g., a ternary relationship is required to capture the time-based information between a sensor (*thing*) and an observation. The three concepts involved in this ternary relationship are associated as follows - a 'Sensor Observation' is generated by a 'Sensor' at some 'TimeInstant'. This temporal dynamicity in data is relevant in several domains such as text processing, time series analysis, and sensor networks. In this study, we reflect upon the ontologies that include temporal concepts for capturing dynamic information.

A detailed study on time ontologies is presented by Ermolayev *et al.* [67]. Their study encompasses several time ontologies discussed across several domain areas (until 2014) encompassing almost all communities in the field of computer science. From their study, we identify 4 different facets of time that affect the choice of temporal concepts in IoT domain. These facets are as follows:

1) Choice of time: Time, for defining IoT data collection, is represented as either absolute (a certain UNIX timestamp or a particular date-time) or relative (before or after some other time). These absolute and relative concepts indicate quantitative and qualitative aspects of temporal information respectively.
2) Granularity of time: This refers to slicing the concepts of time as per the required density (such as years, minutes, or seconds). The granularity of time also varies with the temporal structure used to define time. The temporal structure may consist of a single time instant for event-based data collection; or a sequence of time instants known as a time interval for continuous data collection with(out) a time duration and starting and ending time instants. Temporal structure may also include calendar-clock information to represent relative time. Granularity may also be defined in terms of a time unit as per application requirements. For e.g., in an environment sensing application, sensors may be

[11] http://www.qudt.org/
[12] https://www.w3.org/TR/vcard-rdf/

TABLE 4. A comparison of temporal concepts included in some of the time-based ontologies. A ✓indicates that concepts are present while a ✗ indicates the absence of concepts.

Ontology	Year	Parent Ontologies	Choice of time		Temporal Structure					
			Absolute	Relative	Time Instant	Time Interval	Calendar-Clock	Time Unit	Periodicity	Qualitative relations
TimeLine[16]	2007	OWL-Time*	✓	✓	✓	DSE	YMdhms	✗	✓	✓
Hong et al. [65]	2016	OWL-Time*	✗	✓	✓	SE	✗	✗	✗	✓
Cox et al. [66]	2016	OWL-Time*	✓	✓	✓	D	TYMdhms	✓	✗	✓
OWL-Time[15]	2017	-*NA*-	✓	✓+	✓	D	TYMdhms	✓	✗	✓

+ Concepts defined are stubs. No properties are associated with these concepts.
*: Previous version of OWL-Time (2006)
D: Duration, S: Starting time instant, E: Ending time instant
T: Timezone, Y: Year, M: Month, d: Day, h: Hour, m: Month, s: Second

required to collect data every 15 minutes. In this case, 15 minutes is considered as one unit of time.

3) Periodicity: Periodicity indicates whether the sensor data collection is sporadic, periodic, or uncertain. The periodicity of sensor data collection varies largely depending on IoT application area. For e.g., varying periodicity for an environment monitoring app are periodic sensor data collection, uncertain events-based actuations, and sporadic instants/intervals for event-based sensor data collection.
4) Qualitative properties: In order to reason/query upon temporal data, qualitative relations may be required to compare events in qualitative terms. E.g., In case of forest monitoring, a query may request for CO_2 sensor data *after* a fire event was detected. Thus, a time ontology for IoT domain may also consist of qualitative concepts and properties to improve reasoning and querying.

Table 4 summarizes the different time ontologies proposed in the literature with the concepts supported by them. Several ontologies that define temporal context have been proposed, for example, **DAML-Time** [68] (DARPA Agent Markup Language project Time initiative, 2002), **DAML-S**[13] (DAML for Web Services) (2003), and **KSL-Time** [69] (2002) to name a few. DAML-Time focused on concepts to provide a common understanding of time. DAML-S however, provides temporal concepts required to define a web service such as profile, process and time. KSL-Time ontology, on the other hand, provides concepts to distinguish between different types of intervals and granularity. We refrain ourselves from describing them in detail as these ontologies were proposed before 2012.

The most commonly used ontology for defining temporal concepts is the **OWL-Time**[14] ontology. The W3C consortium initially proposed OWL-Time ontology in 2006 for web applications by extending the DAML-Time ontology proposed by Hobbs and Pan [68]. OWL-Time ontology focuses on describing date-time information specified in Gregorian calendar format and conventional clock system. M. Grüninger [70] verified this ontology in an independent study. Over time, new temporal requirements have been realized for systems and the updated version was proposed in 2017[15] which supports some new concepts while deprecating some. The new ontology supports concepts to define absolute time as time instants and intervals; and relative time using a temporal reference system (the concept only is a stub and is not associated with any properties). Time intervals in OWL-Time are expressed using the concept of ''duration'' combined with instances of time instants that define starting and ending points of the time interval. The Gregorian calendar system allows to define a date-time structure for time concepts, and the qualitative properties defined in the ontology support reasoning. A light-weight version of OWL-Time ontology called **Time-Entity** is developed for applications needing only limited concepts [71]. OWL-Time ontology has been used as a base ontology by several other works mentioned earlier like SSN [42] and SmartOntoSensor [43].

Timeline ontology[16] is another time-based ontology that emphasizes on the use of timelines (combinations of multiple time instants and intervals) to denote temporal objects such as signals or videos. It extends the first version of OWL-Time Ontology by providing various concepts for timelines such as 'ContinuousTimeLine', 'DiscreteTimeLine', and 'OriginMap'. Hong *et al.* [65] proposed a temporal ontology which supports reasoning over temporal queries asked in natural language. Their ontology extends the first version of OWL-Time and introduces some additional properties for easy reasoning on natural language queries and generate responses in natural language. Their ontology model proposes the definition of 'interval sets' (multiple time instants and time intervals) that consist of related time instants that help define relative time.

[13] http://www.daml.org/services/daml-s/0.9/
[14] https://www.w3.org/TR/owl-time/
[15] http://w3c.github.io/sdw/time/
[16] http://motools.sourceforge.net/timeline/timeline.html

Instead of using Gregorian calendar for representing time instants, this ontology uses integers and literals to define relative time. Another alternative to represent time instants using non-Gregorian format was proposed by Cox [66]. The author proposed two extensions of the first version of OWL-Time ontology to represent temporal information in non-Gregorian formats. The first extension includes a temporal reference system to allow for relative time instants and time intervals with OWL-Time ontology. The second extension allows a more generalized extension of OWL-Time ontology to include non-Gregorian calendar format.

The temporal concepts proposed by these ontologies are relevant across domains and only include *positions* of time instants/intervals in the time space. Expansion of temporal concepts from locations to events was proposed by Zhang *et al.* [72]. They propose a hierarchical temporal ontology, where the base ontology comprises of concepts to allow description of temporal *events* such as cultural or historical events in Gregorian calendar format and the lower level ontology supports representing time positions (instants and intervals) in Chinese calendar format. As highlighted, these temporal aspects are useful to define the N-ary relationships in IoT systems in multiple ways. As an example, Dey and Dasgupta [35] have demonstrated the use of time ontology to IoT systems. The unified ontologies proposed in IoT literature (such as IoT-Lite [41] and FIESTA-IoT [14]) also borrow concepts from these existing time ontologies to annotate temporal data.

D. ANSWERING "WHO"

In IoT systems, the data collected and the information generated from this data is sensitive in nature. Add to that the large number of physical *things* connected to systems. The availability of such private and sensitive information from such a large number of *things* brings opportunities and challenges. The opportunities include anytime, anywhere access of information, while the challenges include securing *things* and generated information from intruders trying to gain unauthorized access. This problem becomes more challenging when multiple data sources, and heterogeneous protocols and *things* are inter-weaved in systems [73]. Thus, to prevent unauthorized access in IoT systems for security of *things* and information, it is important to define ''who'' are the authorized users/applications with access to diverse pieces of IoT information. Access-control and security mechanisms are commonly used in areas related to databases and information security. Ontologies have been used as a popular tool for representing access-control models across domains owing to the flexibility and extensibility offered by ontologies. The concepts and properties defined in these ontologies help in identifying the rightful users of the right parts of information. In this section, we study the existing access-control ontologies proposed in these domains.

Access control models essentially try to describe the relationships between different users and the information they have access to. The users of the information are known as *subjects* or actors who require access to *resources* which refer to different pieces of information being shared. These resources might vary in terms of their *granularity* based on the subject in consideration. Consider, for example, a smart building where the owner has access to occupancy data of the entire building, while an occupant has access to data of the floor on which s/he resides. Thus, an access control model is described using tuples of the form $< S, R, SR, I >$ where S denotes the subject, R refers to the resource (and its granularity) in consideration, SR defines the relationship between a subject and a resource, and I is used to define additional information on which the relationship may depend.

Depending on application requirements, four types of access control models have been proposed in the literature [79] - mandatory, role-based, context-based, and attribute based. For applications requiring a strict level of control, Mandatory Access Control (MAC) models have been proposed where a central authority classifies subjects and resources to access levels. In order to access some resource, a user must have the same access level or higher. In the example above, for data access with MAC, an occupant of the floor must be assigned similar or higher level of security as the occupancy data of the floor. Since resources in IoT systems are highly dynamic in nature and depend on the context, MAC models are commonly not used. The second type of access control model is Role-Based Access Control (RBAC) where subjects are assigned roles based on levels of information accessible by them. For the smart building application used as an example above, two roles exist - a subject is either the owner or an occupant of the building. The roles assigned to subjects are used to identify the resources accessible by them. The third, and probably the most common type of access control used in IoT systems, is Context-Aware Access Control (CBAC) where the environment is monitored to identify the context. Based on the environment of subjects and resources, access to different granularities of resources are provided to subjects. For example, in the smart building application, all occupants are granted access to the occupancy data in case of a fire. Similar to CBAC, Attribute Based Access Control (ABAC) has been proposed for systems where resources are accessible to subjects based on their attribute values such as a subject's profile or the value of a resource. For more details on different kinds of access control ontologies, we refer the readers to a recent study by Kirrane *et al.* [79].

The access control rules decide whether or not to grant access for a resource (and different granularities) to a subject. Depending on the type of access control used, systems may support static or dynamic access control to resources. As resources in IoT systems (and other domains in general) have multiple users, policy making is also enabled by some systems to allow users to enable new access rules. Some access control models also allow enabling different actions like modifying resources and adding new resources to data sources. Our review of the different access-control ontologies proposed in different domains so far (since 2012) is presented in the text below.

TABLE 5. The different concepts incorporated in access-control ontologies proposed since 2012. MAC (Mandatory Access Control) models are generally not used for IoT systems as they are suitable for strict environments, which is often not the case with dynamic IoT systems. A ✓indicates that concepts are present while a ✗ indicates absence of concepts.

Ontology	Year	Parent Ontologies	Subject	Resource	Resource Granularity	Access Control	Action	Access Rule	Policy making	Type of Access Control	Additional information
PPMO [74]	2012	PPO, WAC, FOAF.	✓	✓	✓	Attribute-based	✓ (Create, Read, Update, Delete)	✓ (Access, NoAccess)	✓	Static	Weights of preferences; conditional operators.
SHIE3LD [75]	2012	SIOC, FOAF, PRISSMA WAC, OWL-Time.	✓	✓	✓	Context-aware	✓ (Create, Read, Update, Delete)	✓	✓	Dynamic	Mobile contexts, access condition sets.
COAC [53]	2013	-*NA*-	✓	✓	✓	Context-aware	✓ (Read, Write)	✓ (Grant, Decline)	✓	Dynamic	Relationship between users; contextual information.
Steyskal et al. [76]	2014	ODRL (v2.0).	✓	✓	✗	Role-based, Attribute-based	✓ (Read, Write, Aggregate, Delete)	✓ (Prohibition, Permission)	✗	Static	Terms of data usage, representing licenses for data use.
Onto-ACM [77]	2014	-*NA*-	✓	✓	✗	Context-aware, role-based	✗	✓ (Permit, Deny)	✗	Dynamic	✗
Daud et al. [78]	2016	-*NA*-	✓	✓	✓	Attribute-based	✗	✓ (Allow, Deny)	✓	Static	✗

Table 5 summarizes the different concepts proposed in access-control ontologies so far. Privacy Preference Ontology (PPO) was initially proposed in 2011 and later extended by Sacco and Breslin [74] as **PPMO** (Privacy Preference Manager Ontology) to include some additional concepts and properties. PPO provides attribute-based access control of resources to subjects (called 'agents') with fine-granularity. The new version also allows subjects to modify data sources by providing them with read and write permissions. It offers an additional feature of associating a weight (priority) with the policy permissions defined. These weights allow access policies with higher weights to bypass access policies with lower weights. Another approach to ensure context-aware access control for systems consisting of mobile devices was proposed by Costabello *et al.* [75]. They proposed a framework for mobile devices - **SHIE3LD** - to allow authorization mechanisms for RDF graph stores. They borrowed concepts from two ontologies - `s4ac` and `prissma`. The unification of these two ontologies offered by SHIE3LD comprises of concepts to identify subjects and their access condition sets (sets of access conditions) based on context of mobile devices. The ontology proposed by Steyskal and Polleres [76] relies on an existing open standard - Open Digital Rights Language (ODRL) - from the field of Linked Data to publish web content. They extend the ODRL standard language and introduce concepts to provide a combination of role-based and attribute-based access-control models. With role-based access control, a subject is an assigner or assignee of a policy, and with attribute-based access control, attributes such as the number of times data is accessed and the time of data access are considered while providing concepts to implement constraints for data access. These constraints are used to restrict subjects (called 'parties') from accessing resources (called 'assets'). However, they do not reflect upon the granularity of resources shared with subjects. Kayes *et al.* [53] proposed a framework with COAC ontology to provide context-aware access control to applications. Their framework provides an upper layer context ontology which is equipped with a lower layer domain-specific policy to define access control rules. The reasoner provided in the framework allows different subjects to access different granularity levels of resources based on their dynamic contexts.

A more generic attribute-based access control ontology was proposed by Imran-Daud *et al.* [78] to provide definition and enforcement of access control policies to large systems with heterogeneous data (such as Online Social Networks and cloud computing). They define an upper layer ontology for ABAC with concepts that can be instantiated to generate domain-specific instances. The generic nature of the ontology makes it easier to define new policies for a larger set of subject and resources.

A more complex context-based access control ontology was proposed by Choi *et al.* [77]. They combine the context of a subject and a resource with his/her role to determine the right access permissions. They propose a generic context ontology which can be reasoned upon by a SPARQL reasoner to dynamically grant access to a resource. To extend services and data access from one domain to another, Mouliswaran *et al.* [80] proposed to convert a matrix containing permissions (rows with subjects and columns with resources) into role-based access control ontology. They propose a methodology to build a static 'inter-domain role-based' access control ontology by identifying domains, roles of subjects in these domains, and the relationships between inter-domain subjects.

The concepts defined in these generic access-control ontologies can be borrowed by the IoT domain to answer the ''who'' aspect of data collection and access. Defining an IoT ontology with these concepts helps provide secure access to data while providing confidentiality, integrity, and authenticity of users and applications.

Nevertheless, with the implementation of new rules and laws in different parts of the world, like GDPR[17] (General Data Protection Regulation) law in Europe, it is now also necessary to protect the collected data and fulfill the requirements for storage and usage of data. Although access control based ontologies provide authentication mechanisms, but consent to access a particular data is not handled by them. **Fatema et al**. [81] proposed a consent ontology that aims to answer ''who is allowed or denied to do some activity on what data''. Their ontology is an extension of PROV-O[18] ontology. In another similar work Pandit *et al.* proposed **GDPRov** ontology [82] for provenance and consent (also using PROV-O as base ontology). Such ontologies can be used by developers to make systems more robust.

E. ANSWERING "HOW"

So far, we have discussed ontologies and conceptual models that talk about the '*W*'s of data collection and data access in an IoT system. We now discuss the ''how'' aspect involved in collecting and accessing data in IoT applications. We discuss the ontologies that propose concepts to facilitate interaction of users and applications with *things*. A formal specification of these interactions also allows *things* to interact with each other which facilitates exchange of data and generation of higher level information. These formal interactions help in deploying new *things* on heterogeneous platforms, which is otherwise a tedious task as platforms have different hardware and software specifications. In the absence of a formal semantics, this would require IoT applications to incorporate multiple platform specific data-structures, semantics, and API usages. Thus, consistent ways need to be defined to support interaction of IoT applications with physical devices and their representation in the virtual world. This has been made possible by combining service-oriented architecture with IoT platforms [85]. This allows defining homogeneous ''services'' which support interaction and provide multiple functionalities (such as contextual data access) for *things* associated with IoT systems. These services provide standardized interfaces to interact and access *things* on heterogeneous platforms.

Service discovery frameworks and service ontologies have been proposed to allow service providers (testbed owners) and service requesters (IoT applications) to interact. For a brief review of these frameworks up to year 2013, we refer the readers to [86]. In an ideal IoT system, service providers or IoT testbeds should advertise the available services and make them available/visible to intended requesters (users/IoT applications). The service requesters, on the other hand, should be able to search for services matching their requirements using search terms. These requirements should be matched to services advertised by providers to provide relevant information to requesters about using services. In this section, we study some of the service ontologies proposed to define concepts to facilitate advertising, discovery, and usage of IoT services.

Table 6 lists the different service models proposed in the literature. Among the ontologies listed, OWL-S[19] was proposed in 2004 as a standard service model for web services. It includes three major concepts:

1) Service Profile: A service profile describes ''what the service does''.[19] The service providers use the concepts defined in a service profile to annotate generic features offered by a service. It includes semantic description of information that allows service requesters to identify a service based on its functionalities and limitations. Thus, a service profile lists the metadata information of a service which helps in identifying a service based on its requirements from a list of services available in some repository.
2) Service Process: It consists of semantic description of how to ask for and use a service. A service process is a specification of the multiple ways in which a user can interact with a service. For enabling this, a service process includes concepts to define Input, Output, Preconditions, and Effects (IOPE) of a service. Inputs and Outputs are used when a process is used to generate new information (combining data from multiple sources to generate abstract information) while preconditions and effects are used to generate a change in the world (perform actuations).
3) Service Grounding: It specifies the details of how to actually use a service. It consists of semantic description of the communication protocol and the message exchange format used, among other information used for exchanging data. Service Grounding acts as a mapping from abstract to concrete specifications of elements required to interact with a service (IOPE).

Most of the IoT frameworks use OWL-S as an upper ontology to define their service models. Nambi *et al.* [9] and De *et al.* [83] use OWL-S as upper ontologies for defining their service models. These ontologies use only some concepts from OWL-S ontology as suited by their framework requirements. Wang *et al.* [84] use OWL-S as an upper ontology to define service concepts for an IoT framework. They focus more on enhancing the service profile by defining concepts for QoS and QoI associated with a service. However, they limit service grounding to two types only - SOAP and WSDL. This restricts their model usage to applications supporting only SOAP and WSDL services and also restricts efficient service discovery. Nambi *et al.* [9] extend the OWL-S service model to include one more component for

[17] https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en
[18] https://www.w3.org/TR/prov-o/
[19] https://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/

TABLE 6. A brief summary of the semantic models comprising of service related concepts. All these ontologies (except OWL-S) use OWL-S as an upper ontology. A ✓indicates the presence of concepts while a ✗ indicates absence of concepts.

Ontology	Year	Parent Ontologies	Service Profile	Service Process	Service Grounding	Service Policy
OWL-S[19]	2004	-*NA*-	✓	✓	✓	✗
De et al. [83]	2012	GeoNames, OWL-S, QU, SSN*.	✗	✓	✗	✗
Wang et al. [84]	2012	OWL-S, SSN*	✓	✓	✓+	✗
CACOnt [54]	2013	OWL-S, FOAF.	✓	✓	✓	✗
Nambi et al. [9]	2014	SSN*, GeoNames, OWL-S.	✓	✓	✓	historic and contextual information
IoT-Lite [41]	2016	SSN*, QU, WGS84.	✓	✓	✗	✗

+ Additional concepts defined for QoS and QoI related to services.
*: First version of SSN (proposed in 2005)

IoT systems - Service Policy. It allows description of policies under which a service can be used - historical information and real-world information (context). Their work presents an abstract level information of this component and lacks an in-depth explanation of the extended service model. This extension to OWL-S service ontology makes it adaptable to dynamic environments monitored in IoT systems.

To the best of our knowledge, the use of service-related semantic models in IoT domain is largely influenced by the OWL-S standard. This standard, initially proposed for web-services, is meant for static services. As highlighted by Nambi *et al.* [9], there is a scope to improve upon this standard to make it adaptable to more dynamic service-oriented architectures.

V. DISCUSSION AND CONCLUSION

Several organizations such as W3C have attempted to standardize ontologies for different domains (e.g., SSN for sensor networks and WGS84 for locations). A similar ontology for the IoT domain has been proposed by OneM2M consortium that aims to define a base ontology called OneM2M[20] for machine-to-machine systems. The base OneM2M ontology comprises a large number of concepts and the documentation enlists methods to integrate it with external ontologies used by other systems. It is claimed that any system that uses base OneM2M ontology becomes OneM2M system compatible. Yet, there are several core concepts, such as access control, that are not included within it. The systems using different vocabularies for the missing core concepts within OneM2M system will again cause heterogeneity and interoperability issues when integrating data. To the best of our knowledge, this is by-far the closest system proposed for solving the heterogeneity and interoperability issue for integrating multiple IoT systems. Still, there is a long way ahead to define standards for the IoT domain.

Several frameworks catering to IoT systems have also been proposed in the literature to overcome interoperability issues using semantic techniques. For example, the service-oriented IoT middleware proposed by Wang *et al.* [87] utilizes semantic aspects of information fusion but fails to include semantic models for other aspects including deployment, data collection, and data interpretation to resolve interoperability issues with other systems. Another conceptual IoT framework proposed by Sezer *et al.* [88] discusses design guidelines and supports semantic web technologies. However, they only use SSN ontology with their ontology artifacts that are not described in detail which makes it difficult for other systems to adapt to it and resolve interoperability. Another framework - FIESTA-IoT - proposed as part of the European Union's Horizon 2020 programme also uses a semantic approach [89]. The framework uses a unified ontology to enable federation that uses common semantics and resolves interoperability issues between multiple IoT testbeds. To ease federation of IoT testbeds that have semantic annotations, FIESTA-IoT ontology uses concepts from well known ontologies such as SSN, IoT-lite, DUL, OWL-Time, WGS84 and QU. It follows the best semantic practices highlighted in the literature. However, as new testbeds[21] are added to the federation, new concept requirements have come up and will continue to evolve FIESTA-IoT ontology with time. Still, the FIESTA-IoT ontology lacks some core concepts identified in this work to answer the competency questions. For example, concepts required to provide access control mechanisms within the federated testbeds are currently missing from the FIESTA-IoT ontology.

Unlike the limitations associated with these existing frameworks, we propose the use of a core ontology built around the concepts identified using 4W1H methodology for multiple frameworks to rely upon. We have shown that multiple generic ontologies exist for different identified categories of core concepts which makes it difficult to identify a standard base ontology. The existing ontologies propose different concepts with different vocabularies, thus making it difficult to align the existing ontologies with one another. With new application domains being proposed, it is important to settle on a core horizontal group of concepts that are extensible and interoperable with domain-specific concepts of IoT applications. As application requirements keep varying with time (such as new requirements recognized by the introduction of

[20] http://www.onem2m.org/technical/onem2m-ontologies

[21] http://fiesta-iot.eu/index.php/fiesta-testbeds/

user centric laws and regulations such as GDPR), ontologies need to be redefined and even unified/comprehensive ontologies need to be updated.

To summarize, we highlight the basic requirements for annotating the horizontal aspects of an IoT system. Our study lays the foundation for ontology developers to propose an extensible base ontology. Our work proposes a methodology for the ontology developers, in the IoT domain, and identifies the core concepts for a future standard IoT ontology. We highlight the need for flexibility and extensibility in the base ontology to allow support for integration of vertical concepts for application-specific IoT systems. Going forward, the ontology developers would need to agree on defining a discerning vocabulary before arriving at a standard terminology for the core concepts determined by our work.

ACKNOWLEDGMENT

The authors would also like to thank the FIESTA-IoT consortium for fruitful discussions.

REFERENCES

[1] C. C. Aggarwal, N. Ashish, and A. Sheth, "The Internet of Things: A survey from the data-centric perspective," in *Managing and Mining Sensor Data*, C. C. Aggarwal, Ed. Boston, MA, USA: Springer, 2013, pp. 383–428.

[2] J. Ye, L. Coyle, S. Dobson, and P. Nixon, "Ontology-based models in pervasive computing systems," *Knowl. Eng. Rev.*, vol. 22, no. 4, pp. 315–347, 2007.

[3] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the Internet of Things: Early progress and back to the future," *Int. J. Semantic Web Inf. Syst.*, vol. 8, no. 1, pp. 1–21, 2012.

[4] I. Szilagyi and P. Wira, "Ontologies and semantic Web for the Internet of Things—A survey," in *Proc. IEEE IECON*, Oct. 2016, pp. 6949–6954.

[5] A. Gyrard, M. Serrano, and G. A. Atemezing, "Semantic Web methodologies, best practices and ontology engineering applied to Internet of Things," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Dec. 2015, pp. 412–417.

[6] M. Ledvinka, B. Kostov, and P. Křemen, "JOPA: Efficient ontology-based information system design," in *Proc. Int. Semantic Web Conf.* Kobe, Japan, Springer, 2016, pp. 156–160.

[7] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[8] A. Gyrard, C. Bonnet, and K. Boudaoud, "Enrich machine-to-machine data with semantic Web technologies for cross-domain applications," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Mar. 2014, pp. 559–564.

[9] S. A. U. Nambi, C. Sarkar, R. V. Prasad, and A. Rahim, "A unified semantic knowledge base for IoT," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Mar. 2014, pp. 575–580.

[10] A. Gyrard, C. Bonnet, and K. Boudaoud, "Domain knowledge interoperability to build the semantic Web of things," in *Proc. W3C Workshop Web Things*, 2014, pp. 1–5.

[11] H. Dibowski and K. Kabitzsch, "Ontology-based device descriptions and device repository for building automation devices," *EURASIP J. Embedded Syst.*, vol. 2011, pp. 3:1–3:17, Jan. 2011.

[12] P. Hirmer, M. Wieland, U. Breitenbücher, and B. Mitschang, "Automated sensor registration, binding and sensor data provisioning," in *Proc. CAiSE Forum*, 2016, pp. 81–88.

[13] A. Gyrard, C. Bonnet, K. Boudaoud, and M. Serrano, "LOV4IoT: A second life for ontology-based domain knowledge to build semantic Web of things applications," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2016, pp. 254–261.

[14] R. Agarwal *et al.*, "Unified IoT ontology to enable interoperability and federation of testbeds," in *Proc. 3rd IEEE World Forum Internet Things*, Dec. 2016, pp. 70–75.

[15] L. Nachabe, M. Girod-Genet, and B. E. Hassan, "Unified data model for wireless sensor network," *IEEE Sensors J.*, vol. 15, no. 7, pp. 3657–3667, Jul. 2015.

[16] B. Yan *et al.*, "An ontology for specifying spatiotemporal scopes in life cycle assessment," in *Proc. Diversity+ + @ ISWC*, 2015, pp. 25–30.

[17] M. Niitsuma, K. Yokoi, and H. Hashimoto, "Describing human-object interaction in intelligent space," in *Proc. 2nd Conf. Hum. Syst. Interact. (HSI)*, 2009, pp. 395–399.

[18] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4W1H in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 42–48, Aug. 2014.

[19] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in Internet of Things: a survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 1–27, Feb. 2018.

[20] M. Chen, J. Wan, and F. Li, "Machine-to-machine communications: Architectures, standards and applications," *KSII Trans. Internet Inf. Syst.*, vol. 6, no. 2, pp. 480–497, 2012.

[21] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT Middleware: A survey on issues and enabling technologies," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 1–20, Feb. 2017.

[22] M. Ma, P. Wang, and C.-H. Chu, "Data management for Internet of Things: Challenges, approaches and opportunities," in *Proc. IEEE Internet Things (iThings/CPSCom) Green Comput. Commun. (GreenCom), IEEE Int. Conf. IEEE Cyber., Phys. Social Comput.*, Aug. 2013, pp. 1144–1151.

[23] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, Apr. 2014.

[24] R. Petrolo, V. Loscri, and N. Mitton, "Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms," *Trans. Emerg. Telecommun. Technol.*, vol. 28, no. 1, p. e2931, 2017.

[25] D. Gil, A. Ferrández, H. Mora-Mora, and J. Peral, "Internet of Things: A review of surveys based on context aware intelligent services," *Sensors*, vol. 16, no. 7, p. 1069, 2016.

[26] A. Karkouch, H. Mousannif, H. Al Moatassime, and T. Noel, "Data quality in Internet of Things: A state-of-the-art survey," *J. Netw. Comput. Appl.*, vol. 73, pp. 57–81, Sep. 2016.

[27] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[28] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.

[29] K. Hashimoto, Y. Yamane, S. Suzuki, M. Takaai, M. Watanabe, and H. Umemoto, "An ontology-based validation approach to resolve conflicts in manufacturing design process," in *Proc. 4th Workshop Linked Data Quality (LDQ)*, vol. 1824, 2017, pp. 107–112.

[30] B. Balaji *et al.*, "Brick: Towards a unified metadata schema for buildings," in *Proc. ACM Int. Conf. Embedded Syst. Energy-Efficient Built Environ. (BuildSys)*, 2016, pp. 41–50.

[31] B. Szász, R. Fleiner, and A. Micsik, "iLOC–building indoor navigation services using linked data," in *Proc. Joint Proc. Posters Demos Track 12th Int. Conf. Semantic Syst. (SEMANTiCS), 1st Int. Workshop Semantic Change Evolving Semantics (SuCCESS)*, 2016.

[32] I.-H. Bae, "An ontology-based approach to ADL recognition in smart homes," *Future Gener. Comput. Syst.*, vol. 33, pp. 32–41, Apr. 2014.

[33] D. Vrandečić, "Ontology evaluation," in *Handbook Ontologies*. Berlin, Germany: Springer, 2009, pp. 293–313.

[34] L. Daniele, F. den Hartog, and J. Roes, "Study on semantic assets for smart appliances interoperability: D-S4: Final report," TNO, The Hague, The Netherlands, Tech. Rep. 2013/01077, 2015.

[35] S. Dey and R. Dasgupta, "Sensor knowledge representation with spatiotemporal annotation: An energy sensor ontology use case," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2014, pp. 455–459.

[36] R. D. Eastman, C. I. Schlenoff, S. B. Balakirsky, and T. H. Hong, "A sensor ontology literature review," NIST, Gaithersburg, MD, USA, Interagency/Internal Rep. 7908, 2013.

[37] Y. Shi, G. Li, X. Zhou, and X. Zhang, "Sensor ontology building in semantic sensor Web," in *Internet Things*. Berlin, Germany: Springer, 2012, pp. 277–284.

[38] L. Xue, Y. Liu, P. Zeng, H. Yu, and Z. Shi, "An ontology based scheme for sensor description in context awareness system," in *Proc. IEEE Int. Conf. Inf. Autom.*, Aug. 2015, pp. 817–820.

[39] P. Hirmer, M. Wieland, U. Breitenbücher, and B. Mitschang, "Dynamic ontology-based sensor binding," in *Proc. 20th East Eur. Conf. Adv. Databases Inf. Syst. (ADBIS)* (Lecture Notes in Computer Science), vol. 9809, J. Pokorný, M. Ivanović, B. Thalheim, and P. Šaloun, Eds. 2016, pp. 323–337.

[40] H. B. Pötter and A. Sztajnberg, "Adapting heterogeneous devices into an IoT context-aware infrastructure," in *Proc. IEEE/ACM 11th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst. (SEAMS)*, May 2016, pp. 64–74.

[41] 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," *Pers. Ubiquitous Comput.*, vol. 21, no. 3, pp. 475–487, Jun. 2017.

[42] M. Compton *et al.*, "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics, Sci., Services Agents World Wide Web*, vol. 17, pp. 25–32, Dec. 2012.

[43] S. Ali, S. Khusro, I. Ullah, A. Khan, and I. Khan, "SmartOntoSensor: Ontology for semantic interpretation of smartphone sensors data for context-aware applications," *J. Sensors*, vol. 2017, Jan. 2017, Art. no. 8790198.

[44] A. Gyrard, S. K. Datta, C. Bonnet, and K. Boudaoud, "Standardizing generic cross-domain applications in Internet of Things," in *Proc. Globecom Workshops (GC Wkshps)*, 2014, pp. 589–594.

[45] D. J. Russomanno, C. Kothari, and O. Thomas, "Sensor ontologies: From shallow to deep models," in *Proc. IEEE 37th Southeastern Symp. Syst. Theory (SSST)*, Mar. 2005, pp. 107–112.

[46] I. Niles and A. Pease, "Towards a standard upper ontology," in *Proc. Int. Conf. Formal Ontol. Inf. Syst.*, 2001, pp. 2–9.

[47] R. Hodgson, P. J. Keller, J. Hodges, and J. Spivak. (Mar. 2014). *QUDT-Quantities, Units, Dimensions and Data Types Ontologies*. USA. [Online]. Available: http://qudt.org

[48] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *Knowl. Eng. Rev.*, vol. 18, no. 3, pp. 197–207, 2003.

[49] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 4, pp. 263–277, 2007.

[50] Ö. Yilmaz and R. C. Erdur, "iConAwa—An intelligent context-aware system," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2907–2918, 2012.

[51] J. Rodríguez, M. Bravo, and R. Guzmán, "Multi-dimensional ontology model to support context-aware systems," in *Proc. Int. Conf. Internet Web Appl. Services*, 2012, pp. 80–89.

[52] J. Attard, S. Scerri, I. Rivera, and S. Handschuh, "Ontology-based situation recognition for context-aware systems," in *Proc. 9th Int. Conf. Semantic Syst.*, 2013, pp. 113–120.

[53] A. Kayes, J. Han, and A. Colman, "An ontology-based approach to context-aware access control for software services," in *Proc. Int. Conf. Web Inf. Syst. Eng.* Berlin, Germany: Springer, 2013, pp. 410–420.

[54] N. Xu, W. Zhang, H. Yang, X. Zhang, and X. Xing, "CACOnt: A ontology-based model for context modeling and reasoning," *Appl. Mech. Mater.*, vols. 347–350, pp. 2304–2310, Aug. 2013.

[55] Q. Gao and X. Dong, "A context-awareness based dynamic personalized hierarchical ontology modeling approach," *Procedia Comput. Sci.*, vol. 94, pp. 380–385, Dec. 2016, doi: 10.1016/j.procs.2016.08.058.

[56] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Proc. Int. Symp. Handheld Ubiquitous Comput.* Berlin, Germany: Springer, 1999, pp. 304–307.

[57] G. Brown, C. Nagel, S. Zlatanova, and T. H. Kolbe, "Modelling 3D topographic space against indoor navigation requirements," in *Progress and New Trends in 3D Geoinformation Sciences*. Berlin, Germany: Springer, 2013, pp. 1–22.

[58] S. I. Kim and H. S. Kim, "Ontology based location reasoning method using smart phone data," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2015, pp. 509–514.

[59] A. H. Celdrán, F. J. G. Clemente, M. G. Pérez, and G. M. Pérez, "SeCoMan: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications," *IEEE Syst. J.*, vol. 10, no. 3, pp. 1111–1124, Sep. 2016.

[60] R. Cacciotti, J. Valach, P. Kuneš, M. Čerňanský, M. Blaško, and P. Křemen, "Monument damage information system (MONDIS): An ontological approach to cultural heritage documentation," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 5, p. W1, Sep. 2013.

[61] Y.-P. Liao and F.-T. Lin, "Place name ambiguities in urban planning domain ontology," in *Proc. KEOD*, 2015, pp. 429–434.

[62] M. N. R. Jang, C. Y. Suhr, and Y. G. Lee, "Developing a the advanced IoT (Internet of Things) technology based on spatial information," in *Proc. Int. Conf. Hum.-Comput. Interact.* Cham, Switzerland: Springer, 2016, pp. 416–419.

[63] D. Brickley, "Basic geo (WGS84 lat/long) vocabulary," *Documento Informal Escrito Colaboración*, p. 52, 2006.

[64] T. Flury, G. Privat, and F. Ramparany, "OWL-based location ontology for context-aware services," in *Proc. Artif. Intell. Mobile Syst. (AIMS)*, 2004, pp. 52–57.

[65] M.-D. Hong, K.-J. Oh, S.-H. Go, and G.-S. Jo, "Temporal ontology representation and reasoning using ordinals and sets for historical events," in *Proc. Asian Conf. Intell. Inf. Database Syst.* Berlin, Germany: Springer, 2016, pp. 75–85.

[66] S. J. Cox, "Time ontology extended for non-Gregorian calendar applications," *Semantic Web*, vol. 7, no. 2, pp. 201–209, 2016.

[67] V. Ermolayev, S. Batsakis, N. Keberle, O. Tatarintseva, and G. Antoniou, "Ontologies of time: Review and trends," *Int. J. Comput. Sci. Appl.*, vol. 11, no. 3, pp. 57–115, 2014.

[68] J. R. Hobbs and F. Pan, "An ontology of time for the semantic web," *ACM Trans. Asian Lang. Inf. Process.*, vol. 3, pp. 66–85, Mar. 2004.

[69] Q. Zhou and R. Fikes, "A reusable time ontology," in *Proc. AAAI Workshop Ontol. Semantic Web*, 2002.

[70] M. Grüninger, "Verification of the OWL-time ontology," in *Proc. Int. Semantic Web Conf.*, vol. 7031. Berlin, Germany: Springer, 2011, pp. 225–240.

[71] F. Pan and J. R. Hobbs. *Documentation for an Entry Sub-Ontology of Time in OWL*. [Online]. Available: https://www.isi.edu/~hobbs/damltime/time-entry-documentation.txt

[72] C. Zhang, C. Cao, Y. Sui, and X. Wu, "A Chinese time ontology for the semantic Web," *Knowl.-Based Syst.*, vol. 24, pp. 1057–1074, Oct. 2011.

[73] M. Abomhara and G. M. Køien, "Security and privacy in the Internet of Things: Current status and open issues," in *Proc. IEEE Int. Conf. Privacy Secur. Mobile Syst. (PRISMS)*, May 2014, pp. 1–8.

[74] O. Sacco and J. G. Breslin, "PPO & PPM 2.0: Extending the privacy preference framework to provide finer-grained access control for the web of data," in *Proc. 8th Int. Conf. Semantic Syst.*, 2012, pp. 80–87.

[75] L. Costabello, S. Villata, and F. Gandon, "Context-aware access control for RDF graph stores," in *Proc. ECAI*, vol. 242, pp. 282–287, Aug. 2012.

[76] S. Steyskal and A. Polleres, "Defining expressive access policies for linked data using the ODRL ontology 2.0," in *Proc. 10th Int. Conf. Semantic Syst.*, 2014, pp. 20–23.

[77] C. Choi, J. Choi, and P. Kim, "Ontology-based access control model for security policy reasoning in cloud computing," *J. Supercomput.*, vol. 67, no. 3, pp. 711–722, 2014.

[78] M. Imran-Daud, D. Sánchez, and A. Viejo, "Ontology-based access control management: Two use cases," in *Proc. 8th Int. Conf. Agents Artif. Intell.*, 2016, pp. 244–249.

[79] S. Kirrane, A. Mileo, and S. Decker, "Access control and the resource description framework: A survey," *Semantic Web*, vol. 8, no. 2, pp. 311–352, 2017.

[80] S. C. Mouliswaran, C. A. Kumar, and C. Chandrasekar, "Inter-domain role based access control using ontology," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, 2015, pp. 2027–2032.

[81] K. Fatema, E. Hadziselimovic, H. J. Pandit, C. Debruyne, D. Lewis, and D. O'Sullivan, "Compliance through informed consent: Semantic based consent permission and data management model," in *PrivOnISWC*. 2017, pp. 60–75.

[82] H. J. Pandit and D. Lewis, "Modelling provenance for GDPR compliance using linked open data vocabularies," in *PrivOnISWC*. 2017.

[83] S. De, T. Elsaleh, P. Barnaghi, and S. Meissner, "An Internet of Things platform for real-world and digital objects," *Scalable Comput., Pract. Exper.*, vol. 13, no. 1, pp. 45–58, 2012.

[84] W. Wang, S. De, R. Toenjes, E. Reetz, and K. Moessner, "A comprehensive ontology for knowledge representation in the Internet of Things," in *Proc. IEEE 11th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Jun. 2012, pp. 1793–1798.

[85] Q. Wei, Z. Jin, L. Li, and G. Li, "Lightweight semantic service modelling for IoT: An environment-based approach," *Int. J. Embedded Syst.*, vol. 8, nos. 2–3, pp. 164–173, 2016.

[86] L. D. Ngan and R. Kanagasabai, "Semantic Web service discovery: State-of-the-art and research challenges," *Pers. Ubiquitous Comput.*, vol. 17, no. 8, pp. 1741–1752, 2013.

[87] F. Wang, L. Hu, J. Hu, J. Zhou, and K. Zhao, "Recent advances in the Internet of Things: Multiple perspectives," *IETE Tech. Rev.*, vol. 34, no. 2, pp. 122–132, 2017.

[88] O. B. Sezer, E. Dogdu, M. Ozbayoglu, and A. Onal, "An extended IoT framework with semantics, big data, and analytics," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 1849–1856.
[89] J. Lanza *et al.*, "Experimentation as a service over semantically interoperable Internet of Things testbeds," *IEEE Access*, vol. 6, no. 1, pp. 51607–51625, 2018.

GARVITA BAJAJ is currently pursuing the Ph.D. degree in computer science (specialization in mobile computing) with the Grit Lab, led by Dr. P, Singh, IIIT Delhi, India. Her research interests span across mobile sensor systems, data analysis, and machine learning. From 2015 to 2016, she was a Visiting Researcher with Inria, Paris, where she was involved in projects, such as Sarathi and FIESTA-IoT. Her works have been published in several conferences and workshops, including the IEEE MDM, the ACM Ubicomp adjunct (HASCA), the ACM Sensys Adjunct (CrowdSenSys), and the IFIP NTMS.

RACHIT AGARWAL received the Ph.D. degree in computer science and telecommunications with the University of Pierre and Marie Curie, Paris, in 2013, with the laboratory situated at Telecom SudParis. He holds a Post-Doctoral/Researcher Engineer position at Inria, Paris, where he is currently with the MiMove Research Team. His research interests mainly span the areas related to ICT, especially relating to Internet of Things (IoT), human mobility aspects, semantic technologies, and network science. He has been involved with several European projects (such as EIT Digital's 3cixty and H2020 FIESTA-IoT) and has also served as the Co-PI of Inria's Sarathi Associate team. He received the 2015 Semantic Web Challenge.

PUSHPENDRA SINGH received the Ph.D. degree in the area of mobile computing from Inria, Rennes, France, in 2004. He was with Portsmouth University, Newcastle University, and Inria-Rocquencourt. He is currently an Associate Professor with the Indraprastha Institute of Information Technology, New Delhi. His primary research interest areas are mobile systems and applications, middleware, and ICT for development. His research is funded by DEiTY, ITRA, DST, DRDO, and CEFIPRA. His work has been successfully transferred to industry in the past, leading to creation of start-ups and new products. His work has been deployed in the field for various projects, including in projects related to national schemes, such as the National Rural Health Mission and the National Rural Employment Guarantee Scheme. He is also a Visvesvaraya Young Faculty Fellow.

NIKOLAOS GEORGANTAS received the Ph.D. degree in electrical and computer engineering from the National Technical University of Athens and the Habilitation degree in computer science from UPMC/Sorbonne University. He was a Co-Founder of Ambientic, a spin-off based on ARLES' research, which developed mobile collaborative applications. He is a Research Scientist with Inria, Paris, formerly with the ARLES and currently with the MiMove Research Team, which he leads. His research interests relate to mobile distributed systems, middleware, ubiquitous computing, service-oriented computing, and self-adaptive systems. He has over 80 publications on these topics. He is currently involved in interoperability and QoS analysis of service and thing choreographies across heterogeneous middleware interaction paradigms, as well as interoperability between online social network services. He has been involved in a large number of European research projects, among which recently H2020 CHOReVOLUTION and FIESTA-IoT, and undertaken work package leader and PI roles.

VALÉRIE ISSARNY holds the Director of Research position at Inria, the French institute for research in Information and Communication Science and Technologies, where she led the ARLES research Team until 2013, investigating distributed software systems leveraging wireless networked devices, with a special emphasis on service-oriented systems. She, in particular, studies middleware solutions easing the development of distributed collaborative services, including mobile services deployed over smartphones and interacting with sensors and actuators. Since 2013, she has been the Scientific Coordinator of the Inria@SiliconValley International Lab, promoting and fostering collaboration between Inria and California universities. She is also coordinating the Inria CityLab Program dedicated to smart cities and promoting citizen engagement; the program is developed in collaboration with CITRIS, University of California at Berkeley, and targets urban-scale experiment in Paris and California cities. Her ongoing projects include Ambiciti on urban pollution monitoring through participatory sensing and crowd sourcing, and SocialBus on a middleware solution enabling interactions across social media to support democratic assembly and collective actions.