Data Models and Contextual Information

16

Suparna De, Wei Wang, and Maria Bermudez-Edo

Contents

Abstract

The Internet of Things (IoT) and its applications emphasize the need for being context-aware to be able to sense the changing environmental conditions and to make use of the rich contextual information for analysis. The huge volume and high-velocity characteristics of IoT data necessitates that representation of IoT data takes into consideration the contextual information at scale during every step of the data processing life cycle, from production to storage, publication, and search. This chapter categorizes and describes the diverse forms of IoT data that are obtained from heterogeneous sensing sources. It also presents a framework for describing and analyzing the different types of contextual information that need to be associated with the IoT data in order to drive context-aware management and intelligent analytics. In addition, mechanisms for storing big IoT data and its contextual information are described, and common search and discovery methods for making IoT data accessible to applications and analysis components are presented.

S. De (✉)
School of Computer Science and Electronic Engineering, University of Surrey, Guildford, Surrey, UK
e-mail: s.de@surrey.ac.uk

W. Wang
Department of Computing, Xi'an Jiaotong-Liverpool University, Suzhou, China
e-mail: wei.wang03@xjtlu.edu.cn

M. Bermudez-Edo
Department of Software Engineering, University of Granada, Granada, Spain
e-mail: mbe@ugr.es

Keywords

IoT data models · IoT contextual information · Streaming data · Location models · IoT data storage · IoT data search · Data marketplace

16.1 Introduction

The emergence of the Internet of Things (IoT) paradigm and the increase in the number of IoT-enabled applications are driven by the deluge of data generated as a result of rapid deployment of low-cost sensors and accompanying advances in communication technologies [1]. Figure 16.1 shows the top sources of data in the IoT, as noted in a market insight report of emerging IoT data sources [2]. According to the report, sensor-enabled smartphones constitute 69% of all data across telecommunications, IT, financial services, insurance, energy, healthcare, public sector, manufacturing, supply chain, and logistics application sectors. This is followed by smart meters which make up around 32% of the IoT data sources. As more devices with intelligent data processing capabilities are being connected to the network, data sources from sensors in transportation logistics, healthcare, and the energy sectors have also begun to emerge.

Practical realization of such a wide array of applications is dependent on efficient and scalable techniques to make the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
S. Ziegler et al. (eds.), *Springer Handbook of Internet of Things*, Springer Handbooks,
https://doi.org/10.1007/978-3-031-39650-2_16

Fig. 16.1 IoT data sources

collected IoT data discoverable and searchable. To realize the capabilities rendered by such applications requires things to understand the environment in which they are situated and to convey this in their communications with others, in order to drive true machine-to-machine (M2M) cooperation and applications [3]. Thus, the supporting IoT infrastructure and middleware platforms should include functionalities to extract the contextual information that is embedded in raw IoT data. This contextual information needs to be "fully understood in IoT applications to guarantee an effective, efficient, scalable, and automated decision-making process" [4].

The life cycle of observation and measurement IoT data, starting from its production, to its analysis has been described in a number of research works [5–7] differently, but the main steps in the data processing chain, from the point at which it is sensed and communicated, to the one at which it is discovered for consumption by IoT applications, can be succinctly described as shown in Fig. 16.2:

Production and Communication: This stage is usually concerned about various means for collecting data from the IoT devices or open Application Programming Interfaces (APIs) and conveying the data over the network. The communication layer enables data to be transported and aggregated.

Annotation: IoT data can be individual points of measurement (atomic data) about a particular phenomenon (feature of interest), streaming data represented as a time series or data arriving in heterogeneous or homogeneous time intervals. Creating high-level abstractions of the raw data is needed to infer insights from the raw data. As part of this process, data and its associated context can be annotated with semantic formalisms in a structured format in order to foster interoperability between heterogeneous data sources. This step makes explicit the association of raw data with its inherent context information, which is crucial for IoT applications to perform reasoning and sophisticated analysis.

Storage: IoT data can be stored in various formats, with the most common ones being Comma-Separated Values (CSV), Extensible Markup Language (XML), and Javascript Object Notation (JSON). It can be stored in centralized or

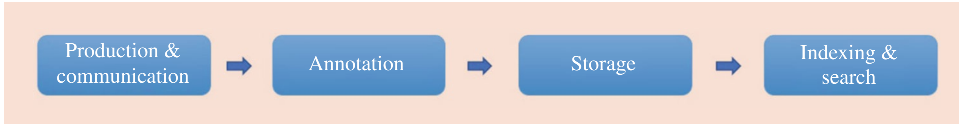

Fig. 16.2 IoT data flow: from its generation to making it searchable

distributed repositories and published in human-readable or machine-processable formats. As noted in [6], the publication step involves decisions regarding whether the data should be published in aggregated form or in its entirety.

Indexing and Search: This step involves mechanisms to enable efficient access to large volume of data, for example, methods for creating and updating the indexing structures in the face of continuous arrival of data streams and making the data available for search. Efficient methods are needed that can reduce the time lag between the arrival of data and its availability, without requiring rebuilding the entire indexing structure. Search mechanisms enable access to the needed data in response to user or application queries, and should support functionalities for searching the data by the associated contextual information such as location, type, or time range.

In line with the above-identified steps of the data life cycle, this chapter outlines the various models and techniques in the state of the art to achieve the requirements in each individual step. Section 16.2 first identifies the various types of IoT data and the accompanying data models and representations. Section 16.3 describes the various context attributes of IoT data, which enable IoT data to be extracted, indexed, and searched. Section 16.4 focuses on the discussion of common effective methods for IoT data storage, publication, indexing, and search mechanisms. Finally, Sect. 16.5 summarizes the chapter on data models and contextual information, and discusses the pertinent issues in modelling the contextual information of IoT data.

16.2 Data Models in the Internet of Things

Data models in the IoT mainly describe Observation and Measurement (O&M) data, mostly focusing on how data is generated, what the data is, and what real-world phenomena or features may be related to the data. In addition, metadata on when and where the data is generated is usually included. This section presents data models designed for annotating both instantaneous and streaming O&M data.

16.2.1 Atomic Data

Atomic data in the context of IoT refers to a piece of indivisible unit of data should its existence be regarded as meaningful to a particular application, for example, a measurement value reported by a chemical sensor, or an image taken by a satellite. Despite the different forms and modalities of IoT data, it is possible to define high-level and abstract models to represent an atomic data item. One of such models that has received considerable consensus from the research community is the linked data model proposed by [8]. There are five attributes in describing atomic IoT data based on semantic knowledge representation formalism:

Location – where the data is generated or reported. The location can be specified with different methods, for example, raw geographical coordinates, geographical area, or Geohash (http://geohash.org/).

Time – when the data is generated or reported, for example, a timestamp.

Type – what the data is about, for example, temperature, humidity, or image.

Value – the actual physical observation and measurement.

Links – additional information about the data, for example, descriptions that provide source or quality of information related attributes.

Figure 16.3 shows the use of the model in describing a sensor observation and measurement data. The five aspects in the data model not only define most of the important properties about an atomic data item but also provide ways on how the data can be used and queried. The model was originally designed for semantic sensor streams when sensors and sensor data were the main focus of the IoT research community. Over the years, the scope of IoT data has been greatly extended; the study by [9] categorizes the basic IoT data to three types: sensor data, text data, and image/video data. Textual data is mostly generated by citizen participatory sensing [10] and has been used in many IoT applications, for instance, traffic analysis [11] and disease tracking [12]. Image and video data have been widely used in smart parking, medical imaging, and public security applications [9].

The linked data model can be easily adapted to represent text, image, and video data. For a sensor reading, the "value" field can be a double or float number with corresponding unit of measurement. For other types of data, the field can be further abstracted to a piece of text (e.g., a twitter message) and an image or video (e.g., represented as a matrix or a sequence of matrices). There are some taxonomies representing quantities, units, dimensions, and values. For example, the QU (Quantities and Units) ontology (https://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu-rec20.html) is

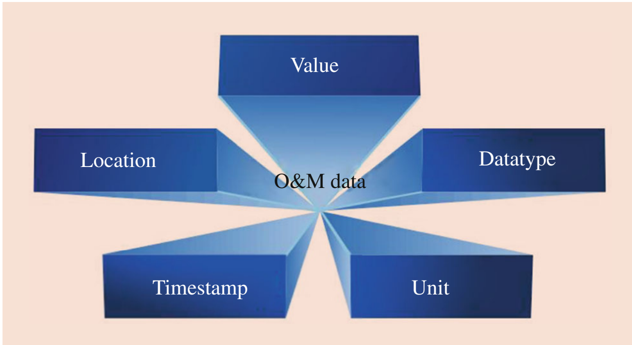

Fig. 16.3 Example of using the atomic data model in describing a sensor observation and measurement data. (Figure adapted from [8])

one of the well-known taxonomies for quantities and units. QU ontology was developed to support different Systems Modelling Language (SysML) users [13].

Time is also represented with different abstractions, and some research efforts have focused on the representation not only of time, but also on some other related concepts such as duration, etc. For example, the time ontology (https://www.w3.org/TR/2017/REC-owl-time-20171019/) is one of the most widely used semantic models to represent time. The concepts included in the time ontology allow us to represent information about duration, temporal position (i.e., date–time information) and topological (ordering) relations. It also includes the vocabulary to express time using different time reference systems, such as Unix time, conventional clock, or geologic time. Time does not only express an instant, but also can represent duration, such as the Gregorian calendar used in [14]. In the latest versions, time ontology has been extended with diverse temporal concepts, including, for example, intervals, instants, and interval relationships. Similarly, the TimeLine ontology [15] represents temporal concepts in conjunction with timeline concepts (e.g., discrete or universal).

The first research works on IoT measurement data modelling focused on the provisioning of the sensor observation data through common interfaces. These works applied Semantic Web technologies to the syntactic XML Open Geospatial Consortium (OGC) schemas for associating the domain knowledge to external models, enabling search queries and cross-domain interpretation. Examples include the work presented in [16], where the sensor data is associated with temporal and geographical concepts from external schemas. In this work, sensor measurements are annotated with time (at which they occurred) and location concepts published by DBpedia [17], although the next steps of data storage and queries over historical measurements are not part of this work. A similar approach of linking sensor data to concepts in DBpedia and GeoNames is presented in the Linked Sensor Middleware (LSM) [18] and in [19].

Subsequent works [20–22] looked at semantic models for sensor data, based on the OGC O&M standard. Prominent among these is the Semantic Sensor Observation Service (SemSOS) O&M model [22] for the weather domain, with the schema including concepts for observations, processes, features (abstraction of real-world entity), and phenomena (property of a feature that can be sensed or measured). The data from sensors installed in weather stations across the US was converted from its raw textual form into Resource Description Framework (RDF) in [20]. The associated model for the data encapsulated the relevant contextual information of the time of observation, its location and type of the observation data. Location context was modelled by linking to relevant concepts in the GeoNames dataset, which allows search queries with approximate location parameters, for example, sensors located “near” a specific place.

Another semantic model is SensorData [21], based on the OGC Sensor Web Enablement (SWE) common data model. Each data record in this model associates the measured quantity with relevant instances from the NASA Semantic Web for Earth and Environmental Terminology (SWEET) ontology [23] to specify its units of measurement. Another model for describing observational data is the SEEK Extensible Observation Ontology (OBOE) [24], which separates the modelling approach for the observations from the entity being observed. This meta-model considers a measurement to consist of a characteristic (i.e., attribute) and a value, with observations in turn consisting of the entity being observed and a set of measurements associated with it. The notion of context represents a relationship between observations.

In contrast to these domain-specific modelling approaches for sensor data, more generic approaches taking into account features of possible IoT smart objects were investigated in some European Framework 7 and Horizon 2020 projects. The Internet of Things-Architecture (IoT-A) project (https://cordis.europa.eu/project/id/257521) in 2010 aimed to define a reference architecture for the IoT. The IoT-A Reference Model [25] established a common base and terminology for IoT architectures and systems. The IoT-A Information sub-model [26] specifically focuses on the IoT information in an IoT system and how it can be defined conceptually through relations and attributes linking the information to the virtual entities, devices, and services in the IoT. In line with the Data-Information-Knowledge-Wisdom (DIKW) hierarchy [27], the IoT-A Information Model defines “information” as something that adds context to data (data represents raw values without useable context). Information is modelled in terms of a value, with associated meta-information such as the time of measurement of the value, the location of measurement, unit of measurement, and other optional elements such as the quality of measurement. The Alliance for the Internet of Things Innovation (AIOTI) initiative’s Working Group 3 (WG03), which focusses on IoT standardization, has proposed a High-Level Architecture (HLA) functional model [28] with two conceptual elements:

the IoT Entity and the App Entity. The IoT Entity is responsible for "thing" representation and performs functions such as data sharing, subscription and notification, device management, location, analytics, and discovery. The App Entity implements application logic. AIOTI considers five key IoT data roles: that of the data provider, which collects data from "things" or from external sources and provides it to IoT data consumers via the IoT data carrier. The other key roles include that of an IoT data application provider which is related to IoT data execution operations such as data preprocessing, analysis, query, and visualization. The IoT data framework provider supports the execution of these data operations by providing the related infrastructure, such as storage, computing resources, and the run time environment. These roles and their interactions are visualized in Fig. 16.4. The mapping of the functional model elements to these data models is also presented in [28], with the IoT Entity performing the data provider and data framework provider roles and the App Entity the application provider and data consumer ones.

The Agent-based COoperating Smart Objects (ACOSO) metamodel [29] borrows the main coarse-grained concepts from the previous models (AIOTI [28], IoT-A [25], and IEEE P2413 [30]). These main concepts include: Virtual and Physical Entity, device, service, and user. Additionally, the ACOSO metamodel includes concepts such as Quality of Service, already used in other models, such as IoT.est or CityPulse [31]. In particular, ACOSO groups the metadata concepts into four categories: Type, which indicates the type of smart object, (e.g., table, wall), from a taxonomy; Device, which defines the physical characteristics of the device; Service, which includes a list of service descriptions for each Smart Object; and Location [32].

On a more concrete level, data measurements generated from IoT smart objects are represented in the Virtual Object (VO) schema [33] that annotates smart object O&M data. This work associates attributes such as a semanticURI and a user-friendly name, to the observation feature type (e.g., temperature), which in turn has properties linking it to external domain models, such as the vocabulary of climate and forecast features (CF). The measurements themselves are encoded to include their numeric values (as a literal), the relevant unit of measurement encoded through links to external schemas such as the Quantities, Units, Dimensions, Values (QUDV) ontology, the time of measurement and the location specified through a Geohash string.

A similar approach of modelling IoT smart objects and their data is adopted in the H2020 Intelligent Knowledge as a Service (iKaaS) project (http://ikaas.com), which investigated IoT data and knowledge processing over multi-cloud environments. In iKaaS, smart objects are abstracted as Virtual Entities (VEs) and measurement data are associated with the output of specific VEs that perform sensing measurements, through the hasMeasurement property in the VE's output function [34]. In addition to the measured value's name, value and type, additional metadata is also associated to the measurement, for instance, the measurement location, accuracy etc. through the hasMeasurementProperty attribute [35].

Readers are directed to a recent survey [36] of modelling efforts for the relevant concepts in the IoT for a comprehensive survey of the relevant literature in semantic modelling for IoT entities.

16.2.2 Streaming Data

A data stream is an ordered sequence of data items which arrives over time at varying time intervals [37]. In the field of IoT, there is no control over the order in which data items arrive. Therefore, it is important to keep a timestamp to each value in the data stream.

As pointed out in [38], much of the earlier work on data streams focused on using sensor Data Stream Management Systems (DSMS) that offer continuous views over sensor data streams through sliding windows defined with specific temporal parameters. Examples include the Global Sensor Networks (GSN) framework [39] that exposes data access APIs to virtual sensors modelled in the GSN system.

In IoT environments, the stream data generated can be voluminous, and some applications need to deal with the data in real time. Furthermore, the devices used in IoT tend to have battery and processing constraints, and the networks used are not always wideband. In such circumstances, lightweight

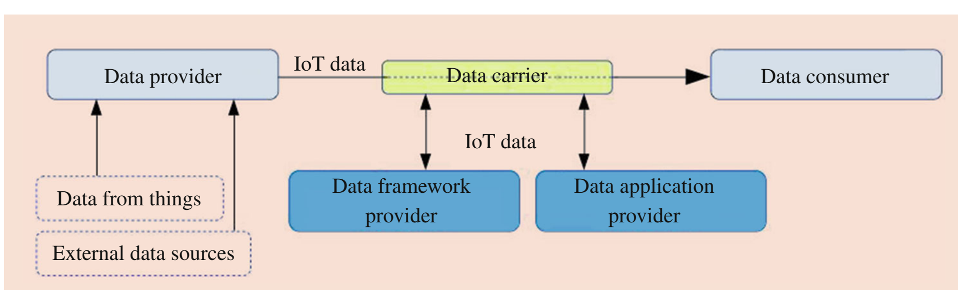

Fig. 16.4 AIOTI IoT data roles. (Adapted from [28])

models are needed in preprocessing steps that could happen in the network edge in order to reduce the processing time and the traffic generated in the transmission of data. This is demonstrated in [1], where various machine learning-enabled microservices are distributed across edge devices, leading to a reduction in the amount of data communicated over the network. With respect to these considerations, the Stream Annotation Ontology (SAO) model proposed by [40] has three main concepts: `StreamData`, `StreamAnalysis`, and `StreamEvent`. The `StreamData` models the data streams, which can be raw data or preprocessed/analyzed data, also in the form of streams. The `StreamAnalysis` annotates the type of preprocessing or analysis performed on the data, and the `StreamEvent` gathers any events inferred from the analysis.

It is important that data model design takes into account user requirements and the purpose of the applications [41]. For example, an application needs to query temperature data in a certain room and to retrieve the values. The useful sensors that provide such information are located, and some applications may need continuous streams of the data. In order to reduce the information sent, the only important information needed to be fetched at that stage is the data value attached to a timestamp. The timestamp is important in order to deal with unordered data, missing data, etc. In that line, the IoT-Stream model proposed by [42, 43] reduces the continuous information by transferring only a two-tuple, value timestamp, which is considered in the concept `StreamObservation` (see Fig. 16.5). With this, the retrieval of stream values, once the source is selected, is just concerned with sending values and timestamps, keeping the queries and responses lightweight.

The RDF Stream Processing Community Group is currently working on the design of a common model for producing, transmitting, and continuously querying semantically annotated streams. The main goal of this group is to extend the data format RDF and the query language SPARQL Protocol and RDF Query Language (SPARQL) for stream data representation and query. The Group has recently published the representation of an RDF stream as a sequence of time-annotated graphs <g [t]>, where g is an RDF graph and t is a timestamp [44]. The main idea behind this extension is based on the previous work of Siemens within the European project Optique [45, 46]. Regarding the extension of SPARQL, the RDF Stream Processing Community Group is studying some of the most representative approaches that address the stream data querying. For example: Instants [47], SPARQLstream [48, 49], EP-SPARQL [50], C-SPARQL [51], CQUEL [52], or STARQL [53], which allow queries not only for stream data, but also static data.

16.2.3 FUTS Data

A recent survey of IoT-enabled cyber-physical-social systems (CPSS) [54] pointed out the diverse range of contextual data sources, taking into consideration not only fixed sensing devices (and hence, locations) but also opportunistic and participatory ones. Opportunistic sensing is mainly concerned about mobile sensing sources that can provide data about the relevant location at a given time point, such as smart city deployments in Santander [55], Madrid [56], Barcelona [57], and China [58], which have sensors mounted on public transport and taxis.

On the other hand, participatory sensing usually involves smart city residents that perform local knowledge gathering through their smartphones enabled with GPS-tracking capabilities and a range of sensors, such as the experiments with portable air quality sensors [59], noise pollution detection in

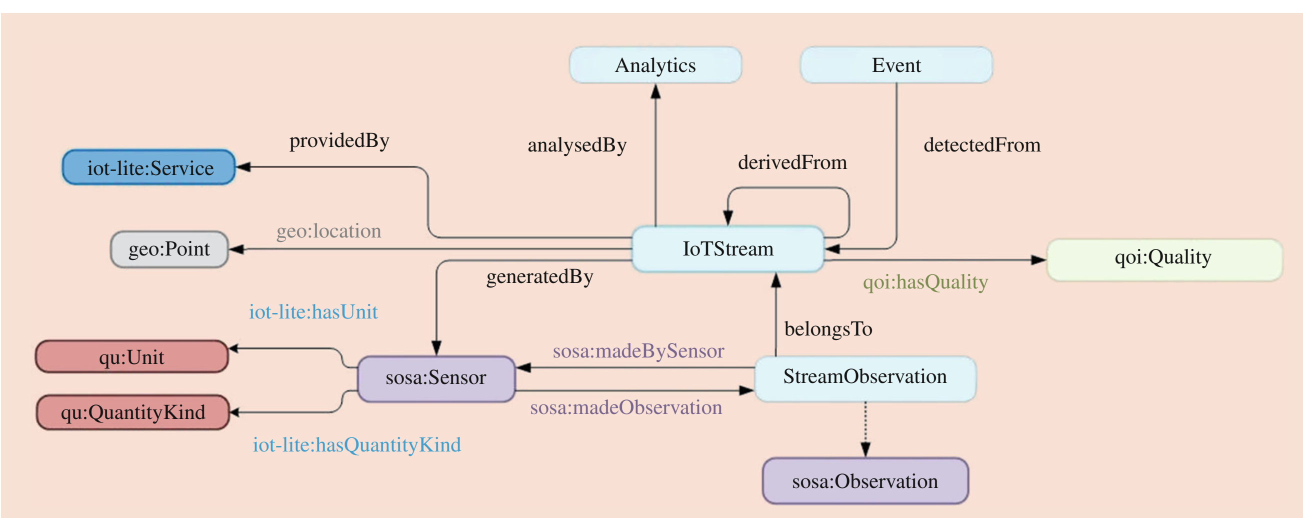

Fig. 16.5 IoT stream ontology. (Adapted from [42])

Fig. 16.6 Example of FUTS data from the SmartSantander city testbed in JSON. (Adapted from [63])


```
{ "id": "3021",
  "latitude": "43.430007",
  "longitude": "–3.949993",
  "title": "bus3021",
  "image": "http://lira.tlmat.unican.es/SmartSantander/iconos/tus.png",
  "content": "<div class='googft-info-window'\n  style='font-family: sans-serif;
font-size: 10px;width: 200px; height: 18em ; overflow-y: auto;'><table
width='100%' border='0'>\n <tr>\n  <td valign='top'>
 <h2 style='color: #5080e1'>NODE 3021</h2>Last update: 2015-01-02
17:33:19<br>Particles: 0.89 mg/m3<br>Humidity: 64.00%
 </td>\n   <td valign='top'></td>\n </tr>\n</table></div>",
  "tags": "BUS"}
{...}
```


Fig. 16.7 Virtual object schema and instance for FUTS data. (Adapted from [63])

ID	3021	**Name**	bus3021
Type	http://dbpedia.org/page/Bus	**Mobile**	Yes
Location {latitude, longitude, geohash}			
{43.430007, –3.949993, eztpn45wn}			
Information {name, value, unit of measurement, time, description}			
{Particles, 0.89, mg/m3, 2015-01-02 17:33:19, density of particles with a diameter between 2.5 and 10 micrometres}			
{Humidity, 0.64, percentage, 2015-01-02 17:33:19, Humidity}			

Melbourne [60], and traffic congestion and incident detection [61]. As noted in a recent survey of IoT data search methods [38], the data streams from mobile sensing sources are not generated at successive, equally time-spaced points, and each observation may be associated with a different geo-location value, thus, differentiating them from pure time-series data. Such resultant IoT data is subsequently termed as "frequently updated, timestamped and structured" (FUTS) data [33, 62]. FUTS data is usually obtainable in a structured data format, such as CSV, XML, or JSON. However, these data models do not necessarily conform to any data standards and the resulting heterogeneous models give rise to compatibility challenges during data storage, search, and retrieval.

An example of a FUTS data is shown in Fig. 16.6, which depicts the observation data values from the Santander city-scale environment monitoring testbed, that monitors temperature, CO, humidity, particles, NO_2, and car presence.

Zhou et al. [63] propose a corresponding data model for such FUTS data, shown in Fig. 16.7, which describes the data sources as 'virtual objects' (VO) with an identifier (ID), name, associated data point in terms of the phenomenon (e.g., humidity), measurement value, unit of measurement, time point, and location (described as a latitude-longitude pair and a Geohash value).

Table 16.1 summarizes the various IoT data modelling efforts for atomic, streaming, and FUTS data. The existing state-of-the-art works are presented along the following dimensions:

IoT data concepts: indicate the elements defined in the model for the data.

Context information: specifies the context metadata associated to the data to describe it.

Application: indicates the application domain to which the model has been applied.

16.3 Context Information in the Internet of Things

Contextual information for the IoT can be loosely defined as the information attached to the Things and their data under consideration that makes them meaningful, discoverable, and useful for reasoning and analytics. Considerable research has

Table 16.1 IoT Data Models for atomic, streaming, and FUTS data

Reference	IoT data concepts	Context information	Application
IoT-A information model [26]	Value, value container	Time, location, unit of measurement, quality of measurement	Monitoring of perishable food in a logistics chain
AIOTI [28]	IoT entity, app entity	Context as relevant for the data roles of provider, framework or application provider, and data consumer	–
ACOSO [29]	Virtual and physical entity, device, service and user	Quality of service, location	Smart University Campus (Smart UniCal)
IoT-Lite [41]	Entity, device, service	Location, time, quantityKind, units, coverage.	FIESTA-IoT (SmartSantander, SmartICS, SoundCity, KETI)
SAO (CityPuse) [40]	StreamData, StreamAnalysis, and StreamEvent	Time, location, quality of information, provenance.	CityPulse (Smart City)
IoT-Stream [43]	IotStream, StreamObservation, analytics and event	Time, location, quality of information, quantityKind, units.	eHealth
O&M linked data model [8]	Value, data type	Timestamp, location (raw geographical coordinates, geographical area, or Geohash), links to external schema	Sensor streams clustering
W. Wang et al. [16]	Sensor observation data	Timestamp, location (Dbpedia)	Weather conditions inference
Linked Sensor Middleware (LSM) [18]	Sensor O&M data	Location (DBpedia and GeoNames)	Sensor data discovery
Barnaghi et al. [19]	Sensor data	Spatial (DBpedia), temporal and thematic	Sense2Web linked sensor data platform
Patni et al. [20]	Measurement data, phenomena, property	Location (WGS84), sampling time, unit of measurement	MesoWest sensor discovery
SensorData [21]	Data record, conditional value, category, vector, geolocation area, position, time	Units of measurement (NASA SWEET ontology), time, location	–
SemSOS O&M model [22]	Observation, process, property, feature, and result data	Location (gml:Point), sampling time, unit of measurement	Weather condition reasoning
SEEK Extensible Observation Ontology (OBOE) [24]	Measurement, value, characteristic, entity, observation, context, relationship	Conceptual contexts such as within, near, overlaps, contains	Annotation and discovery of observational data sets
Zhou et al. [33]	semanticURI, name, value, feature	Unit of measurement (QUDV), time of measurement, location (Geohash)	Sensor data discovery
iKaaS VE model [34]	Measurement, measurement property, name, value, type	Location, accuracy	Environment monitoring, ambient-assisted living, town management
VO model [63]	Name, type, mobility, description, value	Unit of measurement, time, location (latitude, longitude, Geohash)	FUTS data discovery

been conducted in modelling contextual information for IoT-based applications, notably, using ontology-based methods [64]. Most of the existing studies in this line consider common concepts as useful entities in capturing contexts in pervasive computing environments, such as people, location, space, and activities [65, 66]. Some of the approaches also take into account extensibility by allowing more specific entities to be included in context modelling for specific applications, for example, smart home, office, and entertainment [66].

Referring to the data models presented in the previous section, location, time, and additional information (i.e., Links) are considered as main contextual information for the IoT data. Due to the simplicity of time information and the fact that temporal information is always added to IoT observation and measurement data, only semantic modelling of the location and user preference is discussed in detail.

Semantics have been seen as a natural and integrated element for the IoT and are necessarily useful for knowledge representation in IoT applications where a large number of devices and services, and their generated data need to be managed [67, 68]. Without losing much generality, the linked data principles can be used to associate contextual information to what is being modelled [8]. The key advantages include a high degree of reusability and interoperability.

Depending on the nature and requirements of the applications, location can be represented in different forms and granularities; for example, geographical coordinates can be readily obtained via GPS-enabled devices; region or place of interest can be linked to the devices or data which is human understandable; and relative location can be used to represent indoor positioning. To add more semantics, additional information can be specified for a data model. With the linked data, links for additional information can virtually refer to any data objects within a particular domain or the linked open data cloud [69, 70]. In a closed domain, a data model including the contextual entities is usually defined beforehand in an ontology, and instances of the data objects are stored in a knowledge base. With the linked open data cloud, the contextual information can be easily specified by exploiting the massive amount of human knowledge shared and created by millions of people.

16.3.1 Location as Context

Geographical Location

The geographic coordinate system is one of the most widely used system for representation of locations. This system, with its invention dated back to the third century, has been used worldwide. In some cases, it is sufficient to represent only the latitude and longitude; while in others, it could be useful to represent the altitude as well. In the context of IoT, spatial models encompassing the geographic coordinate system are widely used to represent real-world objects' locations [71]. Geographic coordinates also allow to calculate the distance between any two points, and support simple spatial reasoning operations that calculate whether two areas overlap, are adjacent to each other, or one is contained inside the other [72].

There are several standardization bodies working towards a standard for spatial models, such as OGC (http://www.opengeospatial.org/), ISO (https://www.isotc211.org/), and World Wide Web Consortium (W3C) (https://www.w3.org/2015/spatial/wiki/Main_Page), and joining forces, such as the collaboration between OGC and W3C (https://www.w3.org/2015/01/spatial). The Open Geospatial Consortium is an international nonprofit organization committed to make quality open standards for the global geospatial community. The International Organization for Standardization (ISO) is an international standard-setting body composed of representatives from various national standardization organizations. The W3C is the main international standardization organization for the World Wide Web.

The most widely used geographical coordinate models are based on the World Geodetic System 84 (WGS84) location coordinates. WGS84 is the standard U.S. Department of Defense definition of a global reference system for geospatial information. It describes the Earth's size, shape, gravity, and geomagnetic fields, and it is the reference system used by the Global Positioning System (GPS). It is compatible with the International Terrestrial Reference System (ITRS) and uses the International system of units (IS). In every coordinate system, each position on the Earth has a unique coordinate in terms of longitude and latitude. For example, the geo ontology (http://www.georss.org/georss/) originally from OGC and now a joint model of OGC and W3C, defines a point by its long (longitude) and lat (latitude) concepts. The altitude (elev) and some other concepts useful to define areas, such as line, polygon, radius, etc., have also been added.

In IoT, the location of a sensor, which is generally a point, can be usually modelled as a coordinate with longitude, latitude, and maybe altitude. In some other occasions, the coverage of the sensor may be needed, for example, to know whether the data of a sensor is relevant to a particular point in the vicinity of the sensor. Most models today allow specification of the geographic coordinates and coverage, which could be simplified by common shapes such as a circle or a polygon [73, 74].

Another location model is GeoRSS (Reed et al., 2006), http://www.georss.org/, also using WGS-84, that models a basic point with longitude and latitude. The scope of this model is to request, aggregate, share, and map geographically tagged RSS feeds. GeoRSS has two models: the "simple" model and "GML"; the "simple" model supports basic geometries such as point, line, or polygon. GML stands for OpenGIS® Geography Markup Language, a standard from OGC; and is an XML schema for modelling the OpenGIS® Abstract Specification and the ISO 19100 series, with concepts such as coordinate reference systems, geometry, topology, time, units of measure, and generalized values.

As JSON is increasingly used in Internet and replacing XML as a format to transfer information, there is also a specification from the Internet Engineering Task Force (IETF), GeoJson [75], that allows the representation of simple geographical features, along with their nonspatial attributes in JSON. GeoJson represents concepts such as points, line strings, polygons, and multipart collections of these types. Figure 16.8 represents a GeoJSON file, in which one geopoint and two adjacent squares are represented.

GeoJSON has evolved into a lightweight version, TopoJSON, in which the files are reduced by 80% approximately. This reduction is due to the fact that the geometries can share boundaries, avoiding replicating them. For example, in Fig. 16.8, we have represented two adjacent squares in GeoJSON, while in Fig. 16.9 we have represented arcs (line segments) in TopoJSON, which can be accessed by any geometry in the file. In TopoJSON, each square references the index of the arcs it needs. In the example, the arc[0] = arc [−1] is accessed by both squares. TopoJSON also uses an encoding method for file reduction, as well as transformation, consisting of coordinate scaling, translation, and value rounding. All of this eliminates unnecessarily long numbers in coordinates values [76].

There are also some tools that facilitate the use of the spatial models, such as GeoSPARQL [77]. This OGC standard provides a model to represent geospatial data and the

```
{ “type”: “FeatureCollection”,
  “features”: [
    { “type”: “Feature”,
      “geometry”: {“type”: “Point”,
      “coordinates”: [–0.58763,51.24273]},
      “properties”: {
          “name”: “University of Surrey”,
          “postcode”: “GU2 7XH”
          }
      },
      { “type”: “Feature”,
       “geometry”: {
         “type”: “Polygon”,
         “coordinates”: [
           [
               [–0.6,51.1],
               [–0.4,51.1],
               [–0.4,51.3],
               [–0.6,51.3],
               [–0.6,51.1]   ]
           ]
       },
       “properties”: {
         “AreaShape”: “square-left”
         }
       },
    { “type”: “Feature”,
      “geometry”: {
        “type”: “Polygon”,
        “coordinates”: [
               [–0.6,51.1],
               [–0.8,51.1],
               [–0.8,51.3],
               [–0.6,51.3],
               [–0.6,51.1]
            ]
        },
      “properties”: {
          “AreaShape”: “square-right”,
          “VersionOfSquare”: 0.0
        }
      }
    ]
  }
```

Fig. 16.8 Example of a location representation in GeoJson format

ability to query and filter on the relationships between geospatial entities, supporting concepts such as intersects, within, adjacency, etc., that allow search in the spatial space.

Symbolic Location

A symbolic location model gives names to location areas such as region and place of interest, instead of coordinates. Sometimes, it is simpler to use symbolic location names than the geo-coordinates, especially for human beings. For example, it is easier for a user to provide a postal address than to provide the geo-coordinates of that address. There are some models that support these relative notations of locations, for example, GeoNames [78], which is also part of the Linked Data Open Cloud [70]. GeoNames is an open geographical dataset with more than 11 million unique features representing city-scale location features. These features follow a hierarchical categorization. GeoNames matches the geo-coordinates in WGS84 with user-friendly names of places in various languages and consists of other useful information related to those places, such as population, elevation, etc. It also captures the associated contextual information on region containment and distance among locations. As an open data source, users can add or edit names in easy ways. ISO has also a symbolic model (https://www.isotc211.org/) based on addresses including important concepts such as country, city-town, road, etc. The DBPedia dataset [17], which forms part of the Linked Open Data cloud, defines the Place concept for cities and natural environment features (e.g., mountains, rivers, etc.). Each Place concept also delineates with region containment and other spatial relationships. In terms of location

```
{
  “type”: “Topology”,
  “objects”: {
    “UniversityOfSurrey”: {
      “type”: “GeometryCollection”,
      “geometries”: [
        {
          “type”: “Point”,
          “coordinates”: [–0.58763,51.24273],
          “properties”: {
                “name”: “University of Surrey”,
                “postcode”: “GU2 7XH”
          }
        },

        {
          “type”: “Polygon”,
          “properties”: {
         “AreaShape”: “square-left”
          },
          “arcs”: [[0, 1]]
        },
        {
          “type”: “Polygon”,
          “properties”: {
        “AreaShape”: “square-right”,
        “VersionOfSquare”: 0.0
        },
          “arcs”: [[2, –1]]
        }
      ]
    }
  },
  “arcs”: [
      [
          [–0.6,51.3], [–0.6,51.1]
      ],
    [           [–0.6,51.1],
                [–0.4,51.1],
                [–0.4,51.3],
                [–0.6,51.3]   ],
    [
                [–0.6,51.1],
                [–0.8,51.1],
                [–0.8,51.3],
                [–0.6,51.3]   ]
  ]
}
```

Fig. 16.9 Example of a location representation in TopoJson format context, the H2020 TagItSmart project (https://www.tagitsmart.eu) specified its location model [79] to include, in addition to the geo-coordinate information encoded in WGS84 notation, the local county and country information encoded with the NUTS [80] classification system. Figure 16.12 in Sect. 16.3.2 shows an example of a user's location annotated with the TagItSmart location model.

The CityGML OGC standard [81] consists of XML schemas for representing virtual 3D city models. It is based on Geography Markup Language version 3.1.1 (GML3) and represents the city space with concepts for land use, vegetation, tunnels, transportation, water bodies, bridges and buildings, etc.

An important geospatial modelling construct that maps a two-dimensional latitude-longitude pair into a one-dimensional string (consisting of letters and numbers) is the Geohash geocoding system. The Geohash algorithm uses a Base-32 (https://en.wikipedia.org/wiki/Base32) variant and bit interleaving to obtain the string representation of a latitude and longitude pair (the reverse operation is also possible). The encoding represents a hierarchical grid on the map, similar to a Z-order curve, with one bit dividing the entire map area into two halves, with each subsequent bit addition breaking this down into an increasing number of grids, that is, 4, 8, 16, and 32. An example is shown in Fig. 16.10 where the 32 child grids of the geohash string u0 are shown (u00 - u0z), covering parts of France, Germany, Belgium, Switzerland, and northern Italy. The component higher resolution grids of the area covered by the geohash u09 are shown on the right, encoded from u090 to u09z. This area encodes Paris and the region to its south, as shown in Fig. 16.10 below.

As the length of the geohash string increases, the grid size decreases, resulting in higher spatial resolution. The geohash algorithm is in the public domain and the website allows users to input either latitude and longitude or address data to obtain the corresponding geohash string. The hierarchical nature of the underlying grid system and resulting strings mean that nearby locations typically share similar string prefixes, with longer shared prefixes translating to closer geolocations. This property, combined with the one-dimensional string format, finds use in storing point data into a database system, where it is easier to query on a single index rather than on a two-dimensional (latitude-longitude) one and also to conduct proximity search based on string matching (due to the shared prefixes property).

Relative Location

In addition to the outdoor location models reviewed in the preceding sections, a number of research studies have modelled indoor locations to provide spatial context to the IoT entities at a finer granularity. De et al. [83] proposed an indoor location model as part of federated IoT resolution framework, where the main "place" concept can be mapped to indoor constructs such as buildings, floors, commercial/public/residential premises and other structures, with a building modelled as an entity with at least one floor. Also included in the model are various spatial relations to convey adjacency, containment (spatial hierarchy), access, and placement (N, S, E, W direction). Similar approaches were proposed in [84, 85] with resources clustered into different types of rooms, labs, offices, etc., based on semantic relations.

Table 16.2 summarizes the various surveyed works on location-based context models in the IoT. The existing

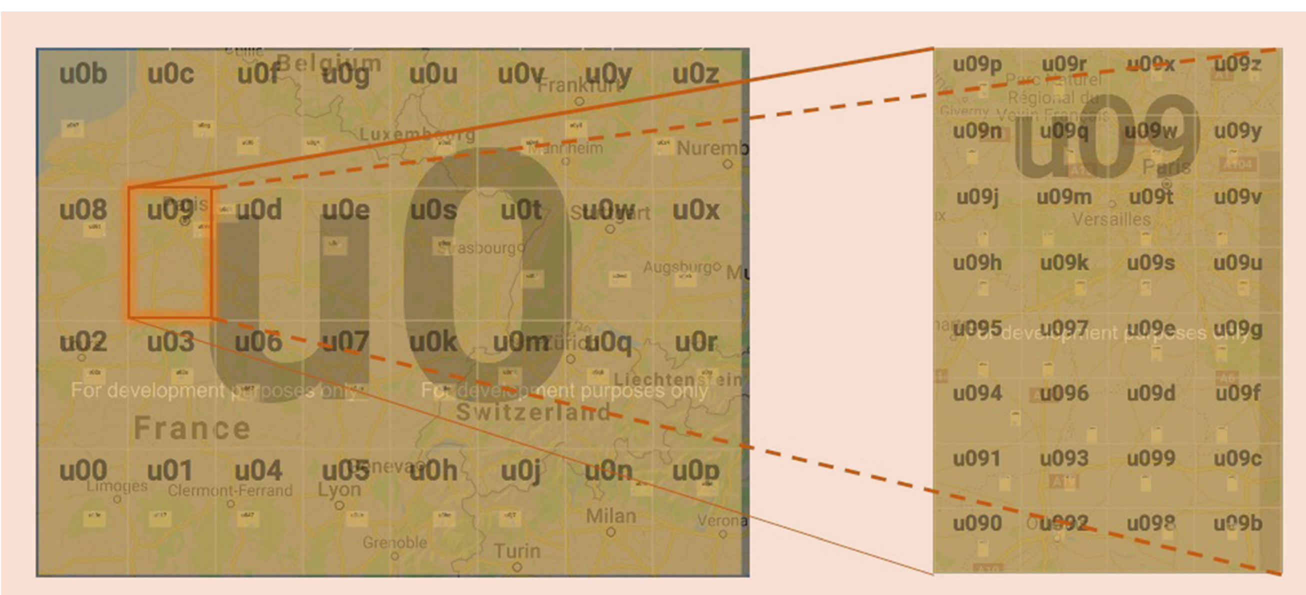

Fig. 16.10 Geohash geocoding system. Maps generated with the Geohash Explorer Service [82]

Table 16.2 Location-based context models

Reference	Modelled concepts	Range	Format	Application
geoRSS ontology (http://www.georss.org/georss/)	Point: long (longitude), lat (latitude), and altitude (elev). line, polygon, coordinate reference systems, geometry, topology, time, units of measure and generalized values, etc.	Global, outdoors	OWL, XML, Relax NG	Any application that needs geolocation
IoT-Lite [41]	Objects, system or resources and services. Point and coverage (polygon, circle, etc.), relative location	Global, outdoors, and indoors	RDF/XML, Turtle, JSON-LD	Smart cities, health
GeoSPARQL [77]	Areas, intersects, within, touches, etc.	Global	RDF	Sensors and actuators (SOSA: Janowicz et al., 2018)
GeoJson [75]	Point, LineString, Polygon, MultiPoint, MultiLineString, and MultiPolygon	Global	JSON	Any application that needs relative location between objects
Geohash geocoding system	Location encoded as a string	Global	Alphanumeric	Geospatial indexing of mobile sensor data and search [63], proximity queries [33]
De et al. [83]	Place (premise, building, room, floor, corridor) and spatial relations	Indoor	OWL	Federated semantic nodes for association analysis [83]
Ben Fredj [84]	Building, room, floor and spatial relations	Indoor	RDF	Clustering of resources based on location
Wang et al. [85]	Place (premise, building, room, floor, corridor) and spatial relations	Indoor	RDF	IoT service annotation and discovery

state-of-the-art works are presented along the following dimensions:

Modelled concepts: indicate the elements defined in the model.

Range: specifies the range of the location model, as either one of indoor/outdoor/global.

Format: specifies the language in which the model has been formalized.

Application: indicates the application domain to which the model has been applied.

16.3.2 User Preferences

User profiles usually store the description of the characteristics of people and their preferences, which can be used as contextual information in many applications. For example, in a travel recommendation application, the ability to drive a car or the health condition of a user can influence the decision about which transport to use, such as car, bicycle, or bus. In the case of an unexpected event, some assumptions could become invalid and the user's profile could again influence the decision.

The early focus of such efforts has been for user interface design [86] and Web information retrieval [87], with more recent efforts leaning towards capture of user profiles and personalization relating to health factors [88–90].

The profile model proposed in [31] includes important concepts such as `personal information`, `interest`, and `abilities` that could enhance the user experience in an application. The personal information concept can annotate aspects such as age, educational level, and employment. The Interest concept covers aspects that users like, such as watching football matches. The Ability concept allows the representation of the activities that the user is able to perform, such as driving a car. Abilities and Interest concepts could be grouped in types (in taxonomies), for easily clustering users (see Fig. 16.11).

The Assisted Living user model [89] defines a number of modules that include a Profile encompassing the person's habits, impairment, and preferences; health encompassing disease, its symptoms and treatment aspects and an Activity recognition module. The user profile in the MobileSage project [88] is aimed at user modelling and personalization reasoning to provide assistance services to people with dementia. The profile aspects are modelled through five concepts: `CapabilityProfile`, `InterestProfile`, `PreferenceProfile`, `EducationProfile`, and `HealthProfile`. The ambient environment is modelled through Context, Location, and Activity classes. The user profile model in [90] facilitates a personalization reasoning mechanism by modelling user preferences in the ontology in terms of preferred media and text size, personal information such as name, age, date of birth, gender, language, and health status, physical and cognitive health conditions and activity. The concept of a user's role has been investigated in the H2020 EU-Japan iKaaS [91] and H2020 TagItSmart project [79], where a user can have different roles in the system that are associated with particular access rights. The user role in these cases is associated with the user's personal information. A combination of location and role specification is used for

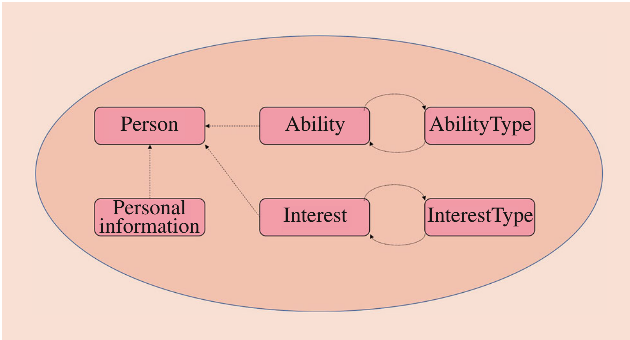

Fig. 16.11 Example of users' profile model

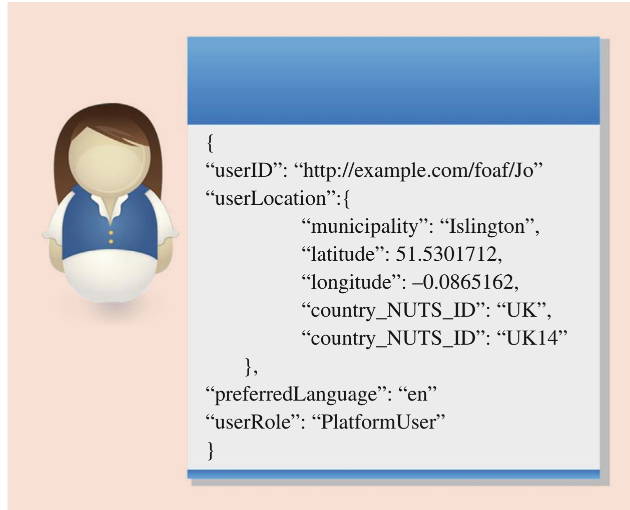

Fig. 16.12 TagItSmart user instance in user repository

delivering different levels of information as well as services to the user, with generic services available to all users, and only certain user roles getting access to specialized platform services.

A key issue in many applications on the Internet of Things is to provide personalized recommendations based on preferences and requirements of the user. An example of a user profile instance, as implemented in the TagItSmart project, is shown in Fig. 16.12, where the user's home location information is used to provide personalized recommendations for local recycling points.

To provide personalized recommendations, a user profile could be stored every time a new user registers in a platform or application. With the user profile, the system could recommend customized actions or personalize the application. However, sometimes, the profiles could be incomplete and the application may not have accurate information about a particular user. In these situations, some techniques could be used to group users in different clusters according to what is present in their profile, assuming that the missing information about a particular user is the same as those users with similar profiles. For example, unsupervised learning algorithms such as clustering can group users based on a set of available variables. The clustering information can be stored in a knowledge base; if a new user only fills in some variables, the user profile system could derive the most suitable cluster for that user and infer the missing information with that of the cluster. The profile information can be updated if new, additional information about the user becomes available.

As for privacy protection, users' profiles could be aggregated into classes (or clusters) and should not contain exact values of profiles attributes that might identify a particular user. For example, ranges of values for age could be used for a user with age between 30 and 50. Likewise, within the knowledge base that stores users' profiles, only the users of a specific application should have their profile accessible by that application, whereas the whole clustered users' profiles (aggregated) can be accessible by any application, as they are anonymized. This clustered knowledge base will contain only clusters which depend on some of the variables, such as young people, student, low or medium income, able to drive, etc. This not only anonymizes the information, but also makes difficult the inference of prejudged statements related to gender, age, religion, or other sensitive information.

16.4 Data Storage and Search in the IoT

16.4.1 IoT Data Storage

The huge volume and high ingestion rate of data in the IoT requires innovative ways for persisting this data and making it accessible for analysis. Since many IoT applications require measurements either as a function of time, the order in which events happened or rates of change of some phenomenon, storing this series of measurements as flat files is not ideal as most retrieval requests would need to access the data based on a time span. For such situations, a time series database (TSDB) is a better choice since it is optimized for queries based on a time range. In TSDBs, facilities are provided to ensure that queries can be efficiently executed for retrieving data from large number of time series for a particular time range. Recent NoSQL approaches, which offer a trade-off against the transactions-level stability, by providing the ability to handle semistructured and de-normalised data, offer greater scalability as well.

In terms of available tools, OpenTSDB [92] is an open source TSDB that uses either Apache HBase or MapR-DB as the underlying storage engine. MapR-DB is a non-relational database that is directly integrated into the file system of MapR distribution derived from Apache Hadoop, enabling rapid ingestion and query of time series data. Since in both HBase and MapR-DB, the number of columns in a database is nearly unbounded, it allows multiple values to be stored in each row (as long as the number of columns with active data in a single row is in the order of a few hundred thousand) [93]. This capability overcomes the limitation of relational databases where one row is used for each measurement. Since data retrieval is dependent on the number of rows that need to be scanned, having multiple measurements in one row enables much faster retrieval times. This capability is known as a wide table design, which is the default feature of OpenTSDB and is also extended with a compressor function that converts wide rows into blobs. A blob is a single data structure which contains the compressed versions of the data in a row. A blob can be highly compressed, so less data needs to be read from disk. So, the time series data is initially inserted into the wide table format and is later compressed into blob structures, facilitating high-performance data recording. OpenTSDB not only features a user interface but also allows direct access to the data via a representational state transfer (REST) interface.

Another industry-standard TSDB is InfluxDB [94], which offers a simple but powerful SQL-like query language and is optimized for query load and data compression. InfluxDB is a schema-less TSDB that includes built-in indexing for string values used to "tag" a measurement. In addition to a command line interface for inserting and querying data (through cURL (https://curl.haxx.se/) scripts), it also offers APIs for programming access. Both OpenTSDB and InfluxDB offer time series retrieval in terms of a number of tag/value pairs, grouping and aggregation functionalities in terms of predefined functions such as minimum, average, and sum. Both of these TSDB tools also feature interfaces that allow integration with Grafana (http://grafana.org/), which is an open source dashboard editor and provides visualization tools for time series data.

Since FUTS data features a different spatial measurement in addition to the temporal one, recent works [63] have extended the InfluxDB TSDB with spatial indices to exploit the spatio-temporal characteristics of FUTS data, with experiments showing impressive data insertion and query performance.

Other widely used, open source tools include the mongoDB NoSQL database, which stores data in JSON-like documents, with useful additional features such as built-in spatial indexing. Cloud platform providers such as Google Cloud Platform include schema-less datastores, for example, Cloud Datastore [95] and Firebase Realtime Database [96] and Amazon's DynamoDB [97].

16.4.2 Data Publication and Subscription

The service-oriented paradigm has also been applied in IoT application development, in which data is not accessed and used with the explicit use of databases, but through a service publication-subscription scheme. Functionalities of a physical device can be abstracted as a virtual service(s) [98, 99], which is essentially similar to a standard Web service. They define input and output, which can be accessed through a service endpoint, and can participate in a service composition process in order to automatically create applications with sophisticated functions [100]. The main differences to Web services are that such services are usually not reliable as they are exposed by IoT devices that mostly operate in dynamic environments.

In this scheme, the services are described according to some semantic models (adding the contextual information about the service, for example, the associated device, input, output, location, and service type) and the service descriptions are stored in distributed semantic repositories [101]. This is often referred to as the service publication. To access data generated by a device, the service(s) that provide the needed data needs to be discovered first according to user's search criteria [101]. The search criteria are defined by the users who specify whatever contextual information they regard as relevant. Once discovered, an application can subscribe to that service for future data communications. It is often the case that there is more than one service returned by the discovery process. In applications that need automatic service composition, the discovered services need to be ranked [102], so that the "best" service can participate in the automatic service composition [100].

16.4.3 Data Search

Indexing

Due to the distributed and often ad hoc nature of IoT data sources, special data indexing methods are often used to constrain the search space for IoT applications. These indexing methods organize search key values and object addresses into catalogues for efficient lookup [38]. Recent studies on IoT indexing and search methods [38, 103] have pointed out the limitations of existing Web-application focused methods in terms of centralized indexing which relies on preexisting links between resources [104, 105] or even distributed indexing which lacks support for index update [106, 107]. IoT data-focused indexing structures need to be adaptive to the high ingestion rate of data, which could be in either the spatial or the temporal dimensions, and could possibly even exceed the rate of user or application queries [103]. This makes the selection of the indexing criteria very important.

Recent surveys [6, 38] have primarily classified IoT indexing techniques into text (or thematic)-based and spatially oriented ones, as depicted in Fig. 16.13.

Text-Based Indexing These indexing methods rely on the textual descriptions of the IoT data in accordance with some

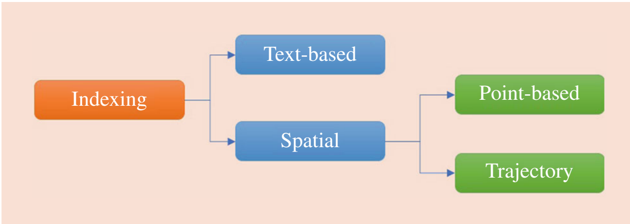

Fig. 16.13 Categorization of IoT data indexing methods

semantic models. Values for some or all of the attributes in the semantic models are used as the keywords for indexing and subsequent search. Notable examples of textual indexing include GSN [108], SenseWeb [109], Dyser [110], Microsearch [111], and IoT-SVK [112]. These approaches can only support low search precision [38]. In some instances [113], XML descriptions of the attributes are used to create a distributed index, with each single indexing structure handling one attribute. However, this limits the search to exact queries only and does not provide support for multi-attribute queries. Similar limitations are experienced with the use of tree indexing, such as the B+ tree with RDF descriptions for IoT data [114].

Spatial Indexing Spatial indexing-based methods typically make use of tree-like structures, such as R-Tree [112, 115, 116], B-tree [117], and quad-tree [103]. These approaches index the Minimum Bounding Rectangles (MBR) of the location range of a node. However, using traditional tree structures for indexing is computational expensive in situations where there are frequent data updates, due to the need for restoring the balance of the tree. Recent approaches have utilized one-dimensional representations of the spatial information, in terms of geohashes with space-filling curves such as Z-order curve [63] to handle the frequent updates of FUTS data.

Trajectory Indexing The survey of indexing approaches in [63] also includes a category of trajectory indexing [118, 119] that involves approximating the trajectories of moving objects to perform faster indexing. However, this requires the time series to have the same length, which is not feasible for many practical IoT deployments. Trajectory approaches also focus on the objects themselves and do not provide means of accessing the generated data. Moreover, time range queries and data aggregation functionalities are not supported, making answering time-dependent queries difficult.

In addition, Fathy et al. [6] propose a time-series based category of indexing methods. Readers are directed to recent surveys [6, 117] of indexing approaches of IoT data for a comprehensive review of the state of the art.

Query Processing and Search

Semantic search is one of the general approaches for search and retrieval of information objects according to users' requirements based on semantic Web technologies [38]. In IoT, the objects and data of interest may be of different types, for example, atomic data, data stream, or service. All these objects should have been semantically described or annotated before search, for example using the data, service, and contextual models presented earlier. The semantic descriptions, or metadata are usually stored in distributed semantic repositories, or as linked data available in the cloud. Various techniques developed for the semantic Web, for example, RDF, SPARQL and linked data, can be leveraged to facilitate the search process and provide accurate and unambiguous results. A common practice is to store semantic descriptions of data and services in RDF format in semantic repositories as such data is usually not updated frequently, while the data itself is stored or archived in distributed streaming databases. Metadata stored in the semantic repositories contains references to the actual data in databases [99, 101].

Due to the heterogeneity of the IoT applications, semantic search should be capable of answering queries of different kinds with high degree of flexibility for users. A search system should offer an easy-to-use interface to help users construct their queries based on the data itself as well as the contextual information that the users are concerned about. The study in [8] listed some of the popular queries that a semantic search system should support:

- Exact queries – where the values for the key attributes are known, e.g., Type, Location, or Time are clearly defined for a requested data item.
- Proximate queries – where the approximate value for the key attributes is known.
- Range queries – where a range of values for the key attributes is known.

Based on this, the system developed in [33] accepts three types of queries for FUTS data collected from mobile sensors:

- Range queries – users can specify a rectangular area on a map, along with the desired time window.
- Distance queries – users can define a circular area of interest by specifying a point on the map and a radius within which observations should have been stored. Time window is also supported.
- Time window and aggregation – in addition to specifying the time range for observations, users also can specify several aggregation functions: minimum, maximum, or average.

The queries are internally translated to a format in line with the semantic description models for data and its context. For example, the system in [33] allows users to specify the location and time with a simple, map-based interface and then translates the input data into a SPARQL query. By leveraging the built-in reasoning capabilities of the SPARQL language, a semantic search system can provide unambiguous results while offering a certain degree of flexibility for end users.

Readers are directed to recent surveys [6, 38, 120–122] of IoT discovery and search mechanisms for a comprehensive review of the state of the art.

Federated Search

Federated search methods [123–126] that take advantage of the established relations between concepts in datasets that are part of the Linked Open Data cloud, have been proposed to answer queries for data that may be stored in different datasets. In contrast to data warehousing methods that aggregate the data from various datasets before the query being issued, the above approaches perform distributed query processing at run time by decomposing the issued query into sub-queries. These approaches apply a variety of grouping and join operations in the sub-queries to minimize the number of remote queries to the datasets and thus, reduce the overall network latency for query response. A comprehensive survey of existing federated search techniques over linked datasets is available in [123] and [127].

16.5 Conclusions

In this chapter, we presented some of the representative models for IoT data and the related contextual information in the literature. We demonstrated the importance of the contextual information in semantically describing different types of data that are unique to IoT, for example, atomic, streaming, and FUTS. A good practice in using the contextual information to annotate IoT data is to create links to existing concepts in existing semantic repositories or databases whereas possible, in accordance with the Linked Data principles. This has the potential to promote interoperability among heterogeneous, distributed systems, and enable automated reasoning by utilising the descriptions of the linked concepts, especially in terms of data storage and search. Moreover, the original contextual information could be enriched with the linked data automatically.

Undoubtedly, data models that allow inclusion of contextual modelling are essential to IoT applications of different sizes and scales. A more prominent usage of these models is perhaps to facilitate data analytics for human users and enable applications to respond and adapt their behavior in an automated fashion to changing ambient environments seamlessly. Among others, the following topics relating to IoT data models and contextual information may need further research.

The ownership and provenance of IoT data has become more important with the rise of participatory sensing in various IoT applications [54]. A recent study [128] has proposed the “sensing-as-a-service” ecosystem, which offers a data marketplace where individual data producers can operate as data owners and can control, manage, monetize and share data with various IoT applications acting as consumers of this data. According to an AIOTI WG white paper on market drivers for IoT-enabled data marketplaces [129], transactions based on IoT data are set to further augment the possibilities of the Data-as-a-Service (DaaS) paradigm by monetizing IoT data. The market drivers are identified as innovation potentially brought about by cross-domain data, where data is produced both internally within an industry (e.g., road infrastructure) and available in real time from external sources (e.g., cars using the roads and reporting on bumps and potholes on the road). A proposed HLA for data marketplace [129] consists of various functions such as `data aggregators` that fuse multiple data streams and provide semantic annotation and contractual terms of use; `data lakes` that store high data volumes with associated metadata to enable data discovery; and `data enrichers` that apply algorithms to yield new insights into the data, providing it as a value-added service.

However, given the penetration of IoT devices into critical industry processes and homes, questions of trust, threats to privacy and confidentiality of commercial intelligence need to be addressed for sustainability of the data marketplace. The AIOTI report [129] on data marketplaces suggests the use of distributed ledger technologies for providing proof of origin for datasets as well as proof of integrity for data lakes. Privacy and security of the data are important concerns in open systems such as the IoT. Specifically, privacy concerns arise from the increasing collection and sharing of personal data, both at an individual as well as at an aggregate level (e.g., GPS traces from individual cars collectively contributing to traffic analysis). Recent PETRAS (privacy, ethics, trust, reliability, acceptability, and security) National Centre of Excellence for IoT Systems Cybersecurity Hub projects such as Displays and Sensors on Smart Campuses (DiSSC) [130] and Resolving Conflicts in Public Spaces (ReCoPS) [131] have demonstrated the benefits of the contextualization of personal data obtained in public spaces through IoT sensors, while also identifying the need for privacy-aware systems to ensure the safe use and adoptability of IoT in public spaces. A survey [132] of the IoT from a data perspective reviews data privacy at both the collection and sharing stages. The authors of this survey list a variety of methods such as spatial delays and the addition of noise to locations for increasing location data privacy. Readers are directed to the work in [133] for a discussion of location privacy-preserving methods.

With regard to the collection and use of personal data, existing methods of ensuring privacy of personal data apply anonymization techniques to aggregated data [134], require a trusted third party or target methods of differential privacy [135], which mediate access to datasets containing personal and sensitive data. However, these approaches are only applicable to aggregated datasets and not practical for continuously generated IoT data streams or implementable in IoT edge devices. The growing penetration of connected and data sensing devices in homes (e.g., smart energy meters, learning thermostats etc.) and participatory sensing in public spaces through smartphones, calls for privacy-preserving systems and analytics approaches that can enable transformation of sensed data at the IoT edge and make it available to IoT applications, without revealing user-identifiable information. Moreover, accidental data disclosure by people and a lack of processes and standards regarding metadata descriptions are some of the key sources of threats to the integrity of IoT data.

With the increasing use of personal devices, especially in the domain of health and smart wearables, annotating and ensuring the Quality of Information (QoI) attribute of the data also gains prominence. There is still no consensus on the dimensions and metrics that can assess the QoI. That lack of consensus could be due to the infancy of the area of research [136]. Trust is one of the most important issues in QoI, specially for crowd-sensing [137], but there are others such as accuracy, precision, completeness, consistency, currentness, reputation, value add, etc. [138–141].

In the IoT environments, the heterogeneity of sources of information and its multi-modal nature results in differences in the QoI [142, 143]; understanding QoI as the utility of the information, or fitness for use [144]. Sensors in IoT are heterogeneous and have differences in precision, accuracy and granularity. Generally, IoT applications make use of different sensors with different QoI. The fusion of such information needs to select the right sensors for each application, depending on the requirements of the application, from the available one and the overlapping information. Selection of the right sensors among multiple ones, or the interpolation methods for missing data because of faulty sensors, also represents a challenge for each individual IoT applications.

Another important challenge is to annotate the spatial component of the QoI. In the IoT environments, multiples sensors could cover a same area. For example, in smart cities, several sensors in different locations could cover the entire city. But the value of the noise sensor will be more accurate near the sensor location, than the one in between two sensors [145]. Furthermore, the accuracy of the spatial component also depends on the propagation model and the spatial infrastructure [146]. For example, temperature is normally similar in a neighbourhood, but noise depends on the buildings infrastructure, and traffic depends on road infrastructures.

The data models reviewed in this chapter are primarily designed for raw IoT data before abstraction and integration. Given the 'big' nature of the IoT data, it has become imperative in many application scenarios that such data needs to be processed at different stages and at different locations. This also accords with the key ideas of a recent proposal of edge computing [147] and distributed intelligence in IoT [148], which has become a popular paradigm for the IoT. It proposes cloud offloading and data processing at the edge of the networks, for example, smart homes or cities. As the result, intermediate data will be generated at the edges. As there are numerous ways to abstract or process data of different kinds, it becomes difficult to understand the meaning of the intermediate data. Often, the abstracted data may be useful, and needs to be reused in many applications. To save the computation cost and facilitate reusability, it is also worth having data models for the abstracted intermediate data.

The need becomes more obvious when data needs to be integrated at the later stage for data analytics. Data integration may happen in different forms, for example, integration of data according to spatial, temporal and type dimensions at sensor networks, gateways or network edges. One form of integration, which is particularly important, happens when data of different modalities co-exists and describes the same event or phenomenon, for example, a natural fire may be described by remote sensor measurement data, social media, or satellite images. The multi-modal data, when integrated together, potentially allows deep insights and actionable knowledge to be obtained, and makes it possible to facilitate building intelligent systems and applications [149]. Considerable research has been conducted in recent years on applying machine learning techniques in processing big IoT and smart city data to design next generation intelligent IoT systems. In particular, deep learning techniques have been widely adopted in processing large amounts of multi-modal data [9]. The research in [150] performed a comprehensive study on the convergence of edge computing and deep learning for IoT. A collaborative end-edge-cloud deep learning computing paradigm is proposed, in which a considerable amount of intermediate data will be generated at end devices, edges and clouds, and transmitted among them. For the sake of optimizing model inputs and narrowing down the searching space for deep learning models, the need for well-defined models for the intermediate data is essential.

References

1. Alturki, B., Reiff-Marganiec, S., Perera, C., De, S.: Exploring the effectiveness of service decomposition in fog computing architecture for the internet of things. IEEE Trans. Sustain. Comput. **7**, 299–312 (2022)
2. Vitria Technology, Real-time Big Data Analytics + Internet of Things (IoT) = Value Creation. In: Market Insights Report, 2015. Available: https://www.vitria.com/wp-content/uploads/2015/01/iot-report-final-011615.pdf

3. Otebolaku, A.M., Lee, G.M.: Towards context classification and reasoning in IoT. In: Proceedings: 2017 14th International Conference on Telecommunications (ConTEL), pp. 147–154 (2017)
4. Kamienski, C.A., Borelli, F.F., Biondi, G.O., Pinheiro, I., Zyrianoff, I.D., Jentsch, M.: Context design and tracking for IoT-based energy management in smart cities. IEEE Internet Things J. **5**(2), 687–695 (2018)
5. Abu-Elkheir, M., Hayajneh, M., Ali, N.A.: Data management for the internet of things: design primitives and solution. Sensors. **13**(11), 15582–15612 (2013)
6. Fathy, Y., Barnaghi, P., Tafazolli, R.: Large-scale indexing, discovery, and ranking for the Internet of Things (IoT). ACM Comput. Surv. **51**(2), 1–53 (2018)
7. Zhou, Y.: Data-driven Cyber-Physical-Social System for Knowledge Discovery in Smart Cities, PhD Thesis, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, UK (2018)
8. Barnaghi, P., Wang, W., Dong, L., Wang, C.: A linked-data model for semantic sensor streams. In: Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing (GREENCOM-ITHINGS-CPSCOM'13), pp. 468–475. IEEE Computer Society, Washington, DC, USA (2013)
9. Chen, Q., et al.: A survey on an emerging area: deep learning for smart city data. IEEE Trans. Emerg. Topics Comput. Intell. **3**(5), 392–410 (2019)
10. Sheth, A.: Citizen sensing, social signals, and enriching human experience. IEEE Internet Comput. **13**(4) (2009)
11. Anantharam, P., Barnaghi, P., Thirunarayan, K., Sheth, A.: Extracting city traffic events from social streams. ACM Trans. Intell. Syst. Technol. **6**(4), 1–27 (2015)
12. Alessio, S., Maria, S.A., Polgreen, P.M.: The use of twitter to track levels of disease activity and public concern in the U.S. during the influenza a H1N1 pandemic. PLoS One. **6**(5), e19467 (2011)
13. OMG System Modeling Language (SysML). (2019, 12 May). Available: https://www.omg.org/spec/SysML/About-SysML/
14. Cox, S.J.: Time ontology extended for non-Gregorian calendar applications. Semant. Web. **7**, 201–209 (2016)
15. Raimond, Y.: A Distributed Music Information System, Ph.D. Thesis, Queen Mary University of London, London, UK (2008)
16. Wang, W., Barnaghi, P.: Semantic annotation and reasoning for sensor data. In: Proceedings of the 4th European Conference on Smart Sensing and Context (EuroSSC2009), Guildford, UK (2009)
17. DBpedia Association. (2019, 25 June). DBpedia. Available: https://wiki.dbpedia.org
18. Le-Phuoc, D., Quoc, H.N.M., Parreira, J.X., Hauswirth, M.: The linked sensor middleware — connecting the real world and the semantic web. In: Proceedings of the Semantic Web Challenge, pp. 42–51 (2011)
19. Barnaghi, P., Presser, M., Moessner, K.: Publishing linked sensor data. In: Proc. 3rd International Workshop on Semantic Sensor Networks (SSN), in Conjunction with the 9th International Semantic Web Conference (ISWC 2010) (2010)
20. Patni, H., Henson, C., Sheth, A.: Linked Sensor Data. In: Proc. International Symposium on Collaborative Technologies and Systems (CTS), pp. 362–370 (2010)
21. Barnaghi, P.M., Meissner, S., Presser, M., Moessner, K.: Sense and sens'ability: semantic data modelling for sensor networks. In: Proceedings of the ICT Mobile Summit (2009)
22. Henson, C., Pschorr, J.K., Sheth, A.P., Thirunarayan, K.: SemSOS: semantic sensor observation service. In: Proc. of the 2009 International Symposium on Collaborative Technologies and Systems (CTS 2009), Baltimore, MD (2009)
23. Raskin, R., Pan, M.: Semantic web for earth and environmental terminology (SWEET). In: Proc. of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data (2003)
24. Bowers, S., Madin, J.S., Schildhauer, M.P.: A conceptual modeling framework for expressing observational data semantics. In: Conceptual Modeling - ER 2008, vol. LNCS 5231, pp. 41–54. Springer-Verlag, Berlin Heidelberg (2008)
25. Bauer, M., Walewski, J.W.: Chapter 3: the IoT architectural reference model as enabler. In: Bassi, A., et al. (eds.) Enabling Things to Talk: Designing IoT Solutions with the IoT Architectural Reference Model, pp. 17–26. Springer (2013)
26. Bauer, M., et al.: Chapter 7: IoT reference model. In: Bassi, A., et al. (eds.) Enabling Things to Talk: Designing IoT Solutions with the IoT Architectural Reference Model, pp. 113–162. Springer (2013)
27. Rowley, J.: The wisdom hierarchy: representations of the DIKW hierarchy. J. Inf. Sci. **33**(2), 163–180 (2007)
28. AIOTI. High Level Architecture (HLA). In: WG03 - loT Standardisation, June 2018, vol. Release 4.0
29. Fortino, G., Russo, W., Savaglio, C., Shen, W., Zhou, M.: Agent-oriented cooperative smart objects: from IoT system design to implementation. IEEE Trans. Syst. Man Cybern. Syst. **48**(11), 1939–1956 (2018)
30. (12 May). IEEE PROJECT P2413—Standard for an Architectural Framework for the Internet of Things (IoT). Available: https://standards.ieee.org/content/ieee-standards/en/standard/2413-2019.html
31. Kolozali, Ş., et al.: Observing the pulse of a city: a smart city framework for real-time discovery, federation, and aggregation of data streams. IEEE Internet Things J. **6**(2), 2651–2668 (2019)
32. Fortino, G., Lackovic, M., Russo, W., Trunfio, P.: A discovery service for smart objects over an agent-based middleware. In: IDCS (2013)
33. Zhou, Y., De, S., Wang, W., Moessner, K.: Enabling query of frequently updated data from mobile sensing sources. In: Proceedings of the 13th IEEE International Conference on Ubiquitous Computing and Communications (IUCC 2014), Chengdu, China (2014)
34. Stavroulaki, V., et al.: D2.1: foundations of semantic data models and tools, IoT and big data integration in multi-cloud environments. In: iKaaS WP2: Technological Foundations I, Public Deliverable (2015) Available: http://ikaas.com/sites/default/files/ikaas/public/content-files/deliverables/iKaaS_D2.1_final.pdf
35. De, S., Georgoulas, S.: D2.2: 1st iKaaS semantic data model, knowledge acquisition and service provision toolbox. In: iKaaS WP2: Technological Foundations I, Public Deliverable (2016) Available: http://ikaas.com/sites/default/files/ikaas/public/content-files/deliverables/ikaaS_D2.2_final.pdf
36. De, S., Zhou, Y., Moessner, K.: Ontologies and context modeling for the web of things. In: Sheng, Q.Z.M., Qin, Y., Yao, L., Benatallah, B. (eds.) Managing the Web of Things, pp. 3–36. Morgan Kaufmann, Cambridge, USA (2017)
37. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: a survey. Inf. Fusion. **37**, 132–156 (2017)
38. Zhou, Y., De, S., Wang, W., Moessner, K.: Search techniques for the web of things: a taxonomy and survey. Sensors. **16**(5) (2016)
39. Aberer, K., Hauswirth, M., Salehi, A.: Infrastructure for data processing in large-scale interconnected sensor networks. In: Proceedings of the 2007 International Conference on Mobile Data Management, Mannheim, Germany (2007)
40. Kolozali, S., Bermudez-Edo, M., Puschmann, D., Ganz, F., Barnaghi, P.: A knowledge-based approach for real-time IoT data stream annotation and processing. In: 2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom), pp. 215–222 (2014)

41. Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., Taylor, K.: IoT-lite: a lightweight semantic model for the internet of things and its use with dynamic semantics. Pers. Ubiquit. Comput. **21**(3), 475–487 (2017)
42. Elsaleh, T., Bermudez-Edo, M., Enshaeifar, S., Acton, S.T., Rezvani, R., Barnaghi, P.: IoT-stream: a lightweight ontology for internet of things data streams. In: Proc. 3rd Global IoT Summit (GIoTS 2019), Aarhus, Denmark (2019)
43. Elsaleh, T., Enshaeifar, S., Rezvani, R., Acton, S.T., Janeiko, V., Bermudez-Edo, M.: IoT-stream: a lightweight ontology for internet of things data streams and its use with data analytics and event detection services. Sensors. **20** (2020)
44. Gutierrez, C., Hurtado, C.A., Vaisman, A.: Introducing time into RDF. IEEE Trans. Knowl. Data Eng. **19**, 207–218 (2006)
45. Kharlamov, E., et al.: Semantic access to streaming and static data at Siemens. J. Web Semant. **44**, 54–74 (2017)
46. Kharlamov, E., et al.: How semantic technologies can enhance data access at Siemens energy. In: ISWC 2014: the Semantic Web, vol. 8796, pp. 601–619. Springer International Publishing, Cham (2014)
47. Rinne, M., Nuutila, E., Törmä, S.: INSTANS: high-performance event processing with standard RDF and SPARQL. In: Proceedings of the 2012th International Conference on Posters & Demonstrations Track, Boston, USA, vol. 914, pp. 101–104 (2012)
48. Calbimonte, J.-P., Corcho, O., Gray, A.J.G.: Enabling ontology-based access to streaming data sources. In: International Semantic Web Conference, ISWC 2010: the Semantic Web, Berlin, Heidelberg, vol. 6496, pp. 96–111. Springer Berlin Heidelberg (2010)
49. Calbimonte J.-P.: Ontology-Based Access to Sensor Data Streams, Ph.D. Thesis, Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain (2013)
50. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: EP-SPARQL: a unified language for event processing and stream reasoning. In: Proceedings of the 20th International Conference on World Wide Web, Hyderabad, India, pp. 635–644. ACM, New York (2011)
51. Barbieri, D.F., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: Querying RDF streams with C-SPARQL. SIGMOD Rec. **39**(1), 20–26 (2010)
52. Le-Phuoc, D., Dao-Tran, M., Xavier Parreira, J., Hauswirth, M.: A native and adaptive approach for unified processing of linked streams and linked data. In: Aroyo, L., et al. (eds.) The Semantic Web – ISWC 2011 Lecture Notes in Computer Science, vol. 7031, pp. 370–388. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
53. Özçep, O., MÖller, R., Neuenstadt, C.: Obda Stream Access Combined with Safe First-Order Temporal Reasoning, Tech. Report. Hamburg University of Technology, Hamburg (2014)
54. De, S., Zhou, Y., Larizgoitia Abad, I., Moessner, K.: Cyber–physical–social frameworks for urban big data systems: a survey. Appl. Sci. **7**(10) (2017)
55. Sanchez, L., et al.: SmartSantander: IoT experimentation over a smart city testbed. Comput. Netw. **61**, 217–238 (2014)
56. Cervigón, P., Gutiérrez, M., Díaz, J., Aránguez, E.: Forecasting for Poaceae pollen with temporal series by applying ARIMA predictive model. In: 7th International Congress Aerobiology, Montebello, QC, Canada (2002)
57. Adler, L.: How Smart City Barcelona Brought the Internet of Things to Life. (2016, 24 June). Available: http://datasmart.ash.harvard.edu/news/article/how-smart-city-barcelona-brought-the-internet-of-things-to-life-789
58. Kang, L., Poslad, S., Wang, W., Li, X., Zhang, Y., Wang, C.: A public transport bus as a flexible mobile smart environment sensing platform for IoT. In: 2016 12th International Conference on Intelligent Environments (IE), pp. 1–8 (2016)
59. Kuznetsov, S., Davis, G., Cheung, J., Paulos, E.: Ceci n'est pas une pipe bombe: authoring urban landscapes with air quality sensors. In: Sigchi Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, pp. 2375–2384 (2011)
60. Jin, J., Gubbi, J., Marusic, S., Palaniswami, M.: An information framework for creating a smart city through internet of things. IEEE Intern. Things J. **1**, 112–121 (2014)
61. Costanzo, A., Faro, A., Giordano, D., Spampinato, C.: Implementing cyber physical social systems for smart cities: a semantic web perspective. In: Proceedings of the 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, pp. 274–275 (2016)
62. Qin, Y., Sheng, Q.Z., Falkner, N.J.G., Dustdar, S., Wang, H., Vasilakos, A.V.: When things matter: a survey on data-centric internet of things. J. Netw. Comput. Appl. **64**, 137–153 (2016)
63. Zhou, Y., De, S., Wang, W., Moessner, K., Palaniswami, S.M.: Spatial indexing for data searching in mobile sensing environments. Sensors. **17**(6), 1427 (2017)
64. Giustozzi, F., Saunier, J., Zanni-Merk, C.: Context modeling for industry 4.0: an ontology-based proposal. Procedia Comput. Sci. **126**, 675–684 (2018)
65. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. Knowl. Eng. Rev. **18**(3), 197–207 (2003)
66. Wang, X., Zhang, D.Q., Gu, T., Pung, H.: Ontology based context modeling and reasoning using OWL. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, pp. 18–22 (2004)
67. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
68. Barnaghi, P., Wang, W., Henson, C., Taylor, K.: Semantics for the Internet of Things: early progress and back to the future. Int. J. Semant Web Inf. Syst. **8**(1), 1–21 (2012)
69. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: a nucleus for a web of open data. In: The Semantic Web, pp. 722–735. Springer, Berlin, Heidelberg (2007)
70. Bizer, C., Heath, T., Idehen, K., Berners-Lee, T.: Linked data on the web (ldow2008). In: WWW '08: Proceeding of the 17th International Conference on World Wide Web, New York, NY, USA, pp. 1265–1266. ACM (2008)
71. Borges, V.: Survey of context information fusion for ubiquitous Internet-of-Things (IoT) systems. Open Comput. Sci. **6**(1) (2016)
72. Becker, C., Dürr, F.: On location models for ubiquitous computing. Pers. Ubiquit. Comput. **9**(1), 20–31 (2005)
73. Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., Taylor, K.: IoT-Lite Ontology. In: W3C Member Submission, 26 November 2015 (2015)
74. Janowicz, K., Haller, A., Cox, S.J.D., Le Phuoc, D., Lefrançois, M.: SOSA: a lightweight ontology for sensors, observations, samples, and actuators. J. Web Semant. **56**, 1–10 (2019)
75. Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., Schaub, T.: The geojson format, RFC 7946 (2016)
76. Orlik, A., Orlikova, L.: Current trends in formats and coordinate transformations of geospatial data — based on MyGeoData converter. Central Eur. J. Geosci. **6**(3), 354–362 (2014)
77. Battle, R., Kolas, D.: Geosparql: enabling a geospatial semantic web. Semant. Web J. **3**(4), 355–370 (2011)
78. Wick, M., Vatant, B.: The geonames geographical database. Available: http://geonames.org. (2012, June 1)
79. D4.1. Resource and Service Modelling Specification. In: TagItSmart Public Deliverable, TagItSmart! Smart Tags driven service platform for enabling ecosystems of connected objects (2017). Available: https://www.tagitsmart.eu/D4.1.pdf
80. NUTS - Nomenclature of territorial units for statistics (2020). Available: https://ec.europa.eu/eurostat/web/nuts/background
81. OGC City Geography Markup Language (CityGML) Encoding Standard, OGC 12-019 (2012)

16

82. Geohashexplorer. (2019, 25 June). Available: http://geohash.gofreerange.com/
83. De, S., Christophe, B., Moessner, K.: Semantic enablers for dynamic digital–physical object associations in a federated node architecture for the Internet of Things. Ad Hoc Netw. **18**, 102–120 (2014)
84. Ben Fredj, S.: Semantic-based middleware for IoT service search, PhD, Laboratoire Traitement et Communication de l'Information, Télécom ParisTech (2014)
85. Wang, W., De, S., Cassar, G., Moessner, K.: Knowledge representation in the internet of things: semantic modelling and its applications. Automatika J. Control Measure. Electron. Comput. Commun. **54**(4), 388–400 (2013)
86. Fischer, G.: User modeling in human computer interaction. User Model. User Adapt. Interact. **11**(1), 65–68 (2001)
87. Kay, J., Kummerfeld, B., Lauder, P.: Personis: a server for user models. In: Proc. Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Malaga, Spain, pp. 201–212 (2002)
88. Skillen, K., Chen, L., Nugent, C.D., Donnelly, M.P., Solheim, I.: A user profile ontology based approach for assisting people with dementia in mobile environments. In: Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC, pp. 6390–6393 (2012)
89. Zografistou, D.: Support for context-aware healthcare in Ambient Assisted Living, MSc. Thesis, University of Crete, Heraklion (2012)
90. Skillen, K.-L., Chen, L., Nugent, C.D., Donnelly, M.P., Burns, W., Solheim, I.: Ontological user modelling and semantic rule-based reasoning for personalisation of Help-On-Demand services in pervasive environments. Futur. Gener. Comput. Syst. **34**, 97–109 (2014)
91. Hashimoto, K., et al.: iKaaS data modeling: a data model for community services and environment monitoring in smart city. In: 2015 IEEE International Conference on Autonomic Computing, pp. 301–306 (2015)
92. OpenTSDB - The Scalable Time Series Database. (2019, 28 June). Available: http://opentsdb.net
93. Dunning, T., Friedman, E., Loukides, M. (eds.): Time Series Databases - New Ways to Store and Access Data. O'Reilly Media (2015)
94. InfluxDB. (2019, 25 June). Available: https://docs.influxdata.com/influxdb/v1.2/
95. Cloud Datastore. (2019, 29 June). Available: https://cloud.google.com/datastore/
96. Firebase Realtime Database. (2019, 29 June). Available: https://firebase.google.com/products/realtime-database
97. Amazon DynamoDB. (2019, 29 June). Available: https://aws.amazon.com/dynamodb/
98. De, S., Barnaghi, P., Bauer, M., Meissner, S.: Service modelling for the Internet of Things. In: Proc. Federated Conference on Computer Science and Information Systems (FedCSIS), Szczecin, Poland, pp. 949–955. IEEE (2011)
99. Wang, W., De, S., Toenjes, R., Reetz, E., Moessner, K.: A comprehensive ontology for knowledge representation in the internet of things. In: Proc. IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Liverpool, UK, pp. 1793–1798. IEEE (2012)
100. Cassar, G., Barnaghi, P., Wang, W., De, S., Moessner, K.: Composition of services in pervasive environments: a divide and conquer approach. In: Proc. IEEE Symposium on Computers and Communications (ISCC), pp. 000226–000232. IEEE (2013)
101. Wang, W., De, S., Cassar, G., Moessner, K.: An experimental study on geospatial indexing for sensor service discovery. Expert Syst. Appl. **42**(7), 3528–3538 (2015)
102. Wang, W., Yao, F., De, S., Moessner, K., Sun, Z.: A ranking method for sensor services based on estimation of service access cost. Inf. Sci. **319**, 1–17 (2015)
103. Fathy, Y., Barnaghi, P., Tafazolli, R.: Distributed spatial indexing for the Internet of Things data management. In: Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 1246–1251 (2017)
104. Shenker, S., Ratnasamy, S., Karp, B., Govindan, R., Estrin, D.: Data-centric storage in sensornets. SIGCOMM Comput. Commun. Rev. **33**(1), 137–142 (2003)
105. Diao, Y., Ganesan, D., Mathur, G., Shenoy, P.J.: Rethinking data Management for Storage-centric Sensor Networks. In: 3nd Biennial Conference on Innovative Data Systems Research (CIDR), vol. 7, pp. 22–31 (2007)
106. Greenstein, B., Ratnasamy, S., Shenker, S., Govindan, R., Estrin, D.: DIFS: a distributed index for features in sensor networks. Ad Hoc Netw. **1**(2), 333–349 (2003)
107. Perera, C., Zasavsky, A., Liu, C.H., Compton, M., Christen, P., Georgakopoulos, D.: Sensor search techniques for sensing as a service architecture for the internet of things. IEEE Sensors J. **14**(2), 406–420 (2014)
108. Aberer, K., Hauswirth, M., Salehi, A.: Infrastructure for data processing in large-scale interconnected sensor networks. In: 2007 International Conference on Mobile Data Management, pp. 198–205 (2007)
109. Kansal, A., Nath, S., Jie, L., Feng, Z.: SenseWeb: an infrastructure for shared sensing. IEEE MultiMed. **14**, 8–13 (2007)
110. Ostermaier, B., Römer, K., Mattern, F., Fahrmair, M., Kellerer, W.: A real-time search engine for the Web of Things. In: Proceedings of the 2010 Internet of Things (IoT), pp. 1–8 (2010)
111. Tan, C.C., Sheng, B., Wang, H., Li, Q.: Microsearch: a search engine for embedded devices used in pervasive computing. ACM Trans. Embed. Comput. Syst. **9**(4), 1–29 (2010)
112. Ding, Z., Chen, Z., Yang, Q.: IoT-SVKSearch: a real-time multimodal search engine mechanism for the internet of things. Int. J. Commun. Syst. **27**(6), 871–897 (2014)
113. Ledlie, J., Ng, C., Holland, D.A.: Provenance-aware sensor data storage. In: Proceedings of the 21st International Conference on Data Engineering Workshops (ICDEW'05), Tokyo, Japan, , vol. 1 (2005)
114. Harth, A., Decker, S.: Optimized index structures for querying rdf from the web. In: Proceedings of the Third Latin American Web Congress (LA-WEB'05), Buenos Aires, Argentina. IEEE (2005)
115. Jirka, S., Bröring, A., Stasch, C.: Discovery mechanisms for the sensor web. Sensors. **9**, 2661–2681 (2009)
116. Bauer, M., Longo, S.: Geographic service discovery for the internet of things. In: Hervás, R., Lee, S., Nugent, C., Bravo, J. (eds.) Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services, pp. 424–431. Springer International Publishing, Cham (2014)
117. Gani, A., Siddiqa, A., Shamshirband, S., Hanum, F.: A survey on indexing techniques for big data: taxonomy and performance evaluation. Knowl. Inf. Syst. **46**(2), 241–284 (2016)
118. Qi, S., Bouros, P., Sacharidis, D., Mamoulis, N.: Efficient point-based trajectory search. In: Advances in Spatial and Temporal Databases, pp. 179–196. Springer International Publishing, Cham (2015)
119. Zhu, Y., Gong, J.: A real-time trajectory indexing method based on MongoDB. In: 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 548–553 (2014)
120. Tran, N.K., Sheng, Q.Z., Babar, M.A., Yao, L.: Searching the web of things: state of the art, challenges, and solutions. ACM Comput. Surv. **50**(4), 1–34 (2017)
121. Pattar, S., Buyya, R., Venugopal, K.R., Iyengar, S.S., Patnaik, L.M.: Searching for the IoT resources: fundamentals, requirements, comprehensive review, and future directions. IEEE Commun. Surv. Tutor. **20**(3), 2101–2132 (2018)
122. Tran, N.K., Sheng, Q.Z., Babar, M.A., Yao, L., Zhang, W.E., Dustdar, S.: Internet of things search engine. Commun. ACM. **62**(7), 66–73 (2019)

123. Zhou, Y., De, S., Moessner, K.: Implementation of federated query processing on Linked Data. In: 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 3553–3557 (2013)
124. Quilitz, B., Leser, U.: Querying distributed RDF data sources with SPARQL. In: Proceedings of the 5th European Semantic Web Conference on the Semantic Web: Research and Applications, Tenerife, Canary Islands, Spain (2008)
125. Schwarte, A., Haase, P., Hose, K., Schenkel, R., Schmidt, M.: FedX: optimization techniques for federated query processing on linked data. In: Aroyo, L., et al. (eds.) Proc. The Semantic Web–ISWC, LNCS, vol. 7031, pp. 601–616. Springer, Heidelberg (2011)
126. Görlitz, O., Staab, S.: SPLENDID: SPARQL endpoint federation exploiting VOID descriptions. In: Presented at the Proceedings of the Second International Conference on Consuming Linked Data, Bonn, Germany (2010)
127. Oguz, D., Ergenc, B., Yin, S., Dikenelli, O., Hameurlain, A.: Federated query processing on linked data: a qualitative survey and open challenges. Knowl. Eng. Rev. **30**(5), 545–563 (2015)
128. Perera, C., Barhamgi, M., De, S., Baarslag, T., Vecchio, M., Choo, K.R.: Designing the sensing as a service ecosystem for the internet of things. IEEE Internet Things Mag. **1**(2), 18–23 (2018)
129. Elloumi, O., Block, T.D., Samovich, N.: Market Drivers and High Level Architecture for IoT-Enabled Data Market Places. In: AIOTI WG Report. AIOTI (2019)
130. Displays and Sensors on Smart Campuses (DiSSC). (2019). Available: https://petras-iot.org/project/displays-and-sensors-on-smart-campuses-dissc/
131. Resolving Conflicts in Public Spaces (ReCoPS). (2019). Available: https://petras-iot.org/project/resolving-conflicts-in-public-spaces-recops/
132. Aggarwal, C.C., Ashish, N., Sheth, A.: The internet of things: a survey from the data-centric perspective. In: Aggarwal, C.C. (ed.) Managing and Mining Sensor Data, pp. 383–428. Springer US, Boston (2013)
133. Aggarwal, C.C., Abdelzaher, T.: Social sensing. In: Aggarwal, C.C. (ed.) Managing and Mining Sensor Data, pp. 237–297. Springer US, Boston (2013)
134. Apple Differential Privacy Team: Learning with privacy at scale. Apple Mach. Learn. J. **1**(8) (2017). Available: https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html
135. Wang, J., Liu, S., Li, Y.: A review of differential privacy in individual data release. Int. J. Distrib. Sens. Netw. **11**(10), 259682 (2015)
136. Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S.: Quality assessment for linked data: a survey. Semant. Web. **7**(1), 63–93 (2016)
137. Restuccia, F., Ghosh, N., Bhattacharjee, S., Das, S.K., Melodia, T.: Quality of Information in Mobile Crowdsensing: Survey and Research Challenges. ACM Trans. Sens. Netw. **13**(4), 34 (2017)
138. Rafique, I., Lew, P., Abbasi, M.Q., Li, Z.: Information quality evaluation framework: extending ISO 25012 data quality model. World Acad. Sci. Eng. Technol. Int. J. Comput. Electr. Autom. Control Inf. Eng. **6**, 568–573 (2012)
139. Natale, D.: Complexity and Data Quality. Presented at the Poster e Atti Conferenza (2015)
140. Bar-Noy, A., et al.: Quality-of-information aware networking for tactical military networks. In: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 2–7 (2011)
141. Thirunarayan, K., Anantharam, P., Henson, C., Sheth, A.: Comparative trust management with applications: Bayesian approaches emphasis. Futur. Gener. Comput. Syst. **31**, 182–199 (2014)
142. Barnaghi, P., Bermudez-Edo, M., Tönjes, R.: Challenges for Quality of Data in Smart Cities. ACM J. Data Inf. Qual. **6**(2–3), 6 (2015)
143. Ahmed, B.S., Bures, M., Frajtak, K., Cerny, T.: Aspects of quality in Internet of Things (IoT) solutions: a systematic mapping study. IEEE Access. **7**, 13758–13780 (2019)
144. Bisdikian, C., Kaplan, L.M., Srivastava, M.B., Thornley, D.J., Verma, D., Young, R.I.: Building principles for a quality of information specification for sensor information. In: 2009 12th International Conference on Information Fusion, pp. 1370–1377 (2009)
145. Bermudez-Edo, M., Barnaghi, P., Moessner, K.: Analysing real world data streams with spatio-temporal correlations: entropy vs. Pearson correlation. Autom. Constr. **88**, 87–100 (2018)
146. Kuemper, D., Iggena, T., Toenjes, R., Pulvermueller, E.: Valid.IoT: a framework for sensor data quality analysis and interpolation. In: Presented at the Proceedings of the 9th ACM Multimedia Systems Conference, Amsterdam, Netherlands (2018). Available: https://doi.org/10.1145/3204949.3204972
147. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)
148. Wang, W., De, S., Zhou, Y., Huang, X., Moessner, K.: Distributed sensor data computing in smart city applications. In: 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–5 (2017)
149. Sheth, A.: Internet of things to smart IoT through semantic, cognitive, and perceptual computing. IEEE Intell. Syst. **31**(2), 108–112 (2016)
150. Wang, X., Han, Y., Leung, V.C.M., Niyato, D., Yan, X., Chen, X.: Convergence of edge computing and deep learning: a comprehensive survey. IEEE Commun. Surv. Tutor. **22** (2020)

Suparna De is a Lecturer in Computer Science at the University of Surrey, UK. She obtained her Ph.D. in Electronic Engineering from the University of Surrey. Her research interests include large-scale data analytics and latent pattern discovery in Web of Things scenarios, social computing, and knowledge engineering methods.

Wei Wang received the Ph.D. degree in computer science from the University of Nottingham, U.K. He is currently a senior associate professor, at the Department of Computing, Xi'an Jiaotong-Liverpool University, China. His research interests include Internet of Things, knowledge discovery, semantic search, and data mining.

16

Maria Bermudez-Edo is an Associate Professor with the University of Granada, Granada, Spain. She received her PhD in Information and Communication Technologies from the same university. Her current research interests include semantic technologies, Internet of Things, and machine learning.