Overcoming the Heterogeneity in the Internet of Things for Smart Cities

Aqeel Kazmi(✉), Zeeshan Jan, Achille Zappa, and Martin Serrano

Insight Centre for Data Analytics, National University of Ireland, Galway, Ireland
{aqeel.kazmi,zeeshan.jan,achille.zappa,martin.serrano}@insight-centre.org
https://www.insight-centre.org

Abstract. In the past few years, the viability of the Internet of Things (IoT) technology has been demonstrated, leading to increased possibilities for novel human-centric services in the smart cities. This development has resulted in numerous approaches being proposed for harnessing IoT for smart city applications. Having received a significant attention by the research community and industry, IoT adaptation has gained momentum. IoT-enabled applications are being rapidly developed in a number of domains such as energy management, waste management, traffic control, mobility, healthcare, ambient assisted living, etc. On the other hand, this high-speed development and adaptation has resulted in the emergence of heterogeneous IoT architectures, standards, middlewares, and applications. This heterogeneity is hindrance in the realization of a much anticipated IoT global eco-system. Hence, the heterogeneity (from hardware level to application level) is a critical issue that needs high-priority and must be resolved as early as possible. In this article, we present and discuss the modelling of heterogeneous IoT data streams in order to overcome the challenge of heterogeneity. The data model is used within the VITAL project which is an open source IoT system of systems. The main objective of the VITAL platform is to enable rapid development of cross-platform and cross-context IoT based applications for smart cities.

Keywords: Internet of Things (IoT) · Interoperability · Linked data · Data model · Smart cities

1 Introduction

With an ever increasing urban population, governments are under pressure to manage resources efficiently while improving human-centric services. Governments and corporations alike are looking for ways to use Information and Communication Technologies (ICT) to provide sustainable solutions to the growing problems originating from rapid urbanization. The Internet of Things (IoT) is seen as a core technology that will help the governments and corporations to manage resources efficiently while improving human-centric services in the modern smart cities. Due to numerous benefits, the Internet of Things (IoT) has

I. Podnar Žarko et al. (Eds.): InterOSS-IoT 2016, LNCS 10218, pp. 20–35, 2017.
DOI: 10.1007/978-3-319-56877-5_2

gained a significant amount of attention from the research community. In addition to research community efforts, serious business decisions taken by numerous major ICT companies such as, Google, Apple, Samsung, and Cisco have transferred the Internet of Things from conceptualization to reality [42]. Today the IoT, with an envisage of 25 billion connected devices by 2020, is seen as a prominent technology that can help in efficient resource management across different sectors such as smart energy management, waste management, smart traffic control, mobility management, smart healthcare, and Ambient-Assisted Living (AAL), etc. [41]. Without doubt, the Internet of Things has paved the way for smart cities to deliver cyber-physical based, context aware, human-centric services.

Attention given by the research community and industry has already led to the development of a number of heterogeneous IoT applications, which are offering services in different domains in the area of smart cities. In IoT, interoperability is seen as the process of integrating different levels of data (generated by an IoT application) that may also use different representation models. IoT systems that generate a set of heterogeneous data streams are unable to communicate with each other at the data level. The work presented in this paper aims to bring interoperability at one common layer by using semantics for storing heterogeneous data streams generated by different IoT eco-systems. This is achieved by means of a common data model using Linked Data technologies. In order to provide interoperability for multiple IoT data streams, we present system agnostic data models that are based on existing ontologies. The data models are used within the so-called Virtualized programmable InTerfAces for innovative cost-effective IoT depLoyments in smart cities (VITAL) project. VITAL uses linked data standards for modelling and accessing data including RDF as a basic data model, JSON-LD as the data format, and ontologies to specify the data in a formal way.

The remainder of the article is structured as follows. First we give a brief overview of the semantic and linked data technologies that are used to develop the data model. Then we present different ontologies, as well as necessary extensions to them for modelling data within VITAL, e.g. for modelling sensors and their measurements, for IoT systems and services, and for Smart City applications. Finally, we conclude the work and present future work plans.

2 IoT Fragmentation

In the recent past, IoT domain has made noteworthy progress across several dimensions, which includes, development of multiple IoT architectures and standards, IoT cloud-enabled middleware platforms, and a large number of IoT deployments in Smart Cities.

2.1 Architectures and Standards

Without doubt, the development of IoT architectures and related technological standards have provided the momentum for deploying scalable, intelligent, and interoperable IoT applications in the modern smart cities. A number of standardization organizations promoted these IoT architectures. Examples include the IoT-A Reference Model (ARM) [2]; the WWW Consortium's (W3C) Semantic Sensor Networks (SSN) incubator group that targeted the context-aware Wireless Sensors [32]; the Research Cluster on the Internet of Things (IERC) which aims to coordinate and build a consensus on ways to realise the IoT vision in Europe [8]; the European Commission's Alliance for Internet of Things Innovation (AIOTI) which aims to assist IoT research and standardization policies [1]; the Open Geospatial Consortium (OGC) which is committed to making open standards for the global geospatial community [18]; and the Electronic Product Code (EPCglobal) which focuses in two areas the RFID and Information Services [6]. The oneM2M standard [26], aims to develop technical specifications that can address the need for a common M2M Service Layer that can be readily embedded within various hardware and software. OM2M [19] is an open source implementation of the oneM2M standard.

2.2 Platforms

A number of middleware platforms have been developed, which facilitate the collection of data from homogeneous and heterogeneous IoT devices. Currently, available platforms provide a range of functionalities e.g. collecting data from hardware sensors, transforming it into a specified representation, and enabling (information access) interfaces for applications. Furthermore, some platforms also use the power of other ICT technologies, notably cloud computing infrastructures, and have opened ways towards IoT/cloud convergence. These platforms encourage the end-users to attach their IoT devices to the cloud infrastructure and enable easy-to-use APIs for retrieving sensor observations and developing applications [14,27]. HomeOS [36] is a platform that provides PC-like abstraction over a wide range of IoT devices. ThingWorx [28] is another IoT platform that facilitates the development of Smart City applications and has already been adopted by many corporations. As part of the IoT/cloud convergence, the OpenIoT (Open Source cloud solution for Internet of Things) is an open-source middleware platform that gathers sensor data and uses the cloud computing mechanism to offer Sensing as a Service [40] model. Examples of other cloud-enabled platforms include the Future Internet of Things (FIT) [9], Xively [33], and Hi-Reply [11] platforms. Similar to VITAL, BIG IoT is another EU project that aims at enabling the emergence of cross-standard, cross-platform, and cross-domain IoT services and applications [3]. However, VITAL differs from the BIG IoT, as BIG IoT does not transform or store the data streams coming from IoT systems.

2.3 Applications

The recent past has witnessed the development of many research and commercial IoT applications, which range from the very basic to more smart and intelligent ones [4,38,39]. These applications are typically developed for specific domains within the scope of modern Smart Cities and often use a large number of IoT devices. Examples of such applications include the BURBA (Bottom Up selection, collection and management of URBAn waste) system an Intelligent Waste management system [7]; the OpenIoT uses cases targeting smart manufacturing and smart agriculture [40]; OnFarm which is an IoT system designed to facilitate smart farming [15]; HeatWatch which is animal monitoring system that keeps track of animal movement [10]; SmartStructures a smart infrastructure monitoring solutions [25], ParkSight a smart parking management system within smart cities [4]; and Echelon which is a smart lighting management system targeting energy efficiency [24]. By looking at these examples, one can imagine that IoT applications are already transforming people's lives while having a significant impact on Smart Cities resource management.

2.4 Smart City Silos

The development of multiple IoT architectures, IoT platforms, and multiple (domain-specific) IoT applications has clearly instigated the momentum of IoT adaptation in Smart Cities. But on the other hand this development has introduced a significantly fragmented IoT landscape resulting from heterogeneity. Apart from development of architectures and platforms, this fragmentation has also resulted from independent IoT deployments, thereby leading to vertical IoT silos. These silos result from technological as well as organizational considerations. Horizontal convergence of these isolated IoT systems is essential to acquire new services and required efficiency [34]. This convergence should not be restricted to the technical integration rather it should also extend to the applications and services spread across different business contexts. IoT Data streams from the different domains should be combined to better manage the city services. For example, real-time information about the traffic flow combined with other information (weather, events, school timings, etc.) can enable traffic prediction based on time scales.

3 VITAL - System of Systems

VITAL has researched for best practices to eliminate the technological and organizational silos in the smart cities. It has introduced an integrated virtualized paradigm for the development, deployment and operation of smart city applications, which emphasizes the collection and processing of data streams from heterogeneous sensors and IoT platforms across the urban environment. This integrated virtualized paradigm is supported by VITAL. VITAL is a system of systems; a system that can support any underlying IoT system. VITAL has taken into account the work performed in the scope of the FP7 IoT-A project and their IoT-A Reference Model (ARM), as a guidance to design the VITAL architecture. The ARM has been used as a source of building blocks (e.g., protocols, interfaces, components), which can be used in order to assemble a concrete architecture. VITAL integrates several of these building blocks, with particular emphasis on

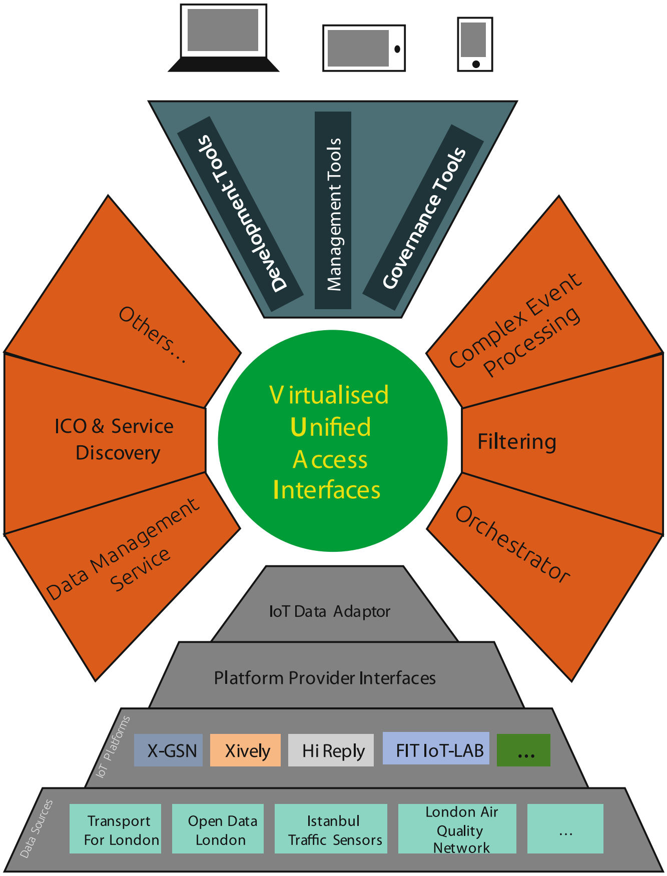

Fig. 1. High-level overview of the VITAL platform.

blocks that deal with services creation, orchestration and protocols, and less emphasis on low-level networking concepts and protocols. The development of VITAL is driven by a number of principles and characteristics:

- `Virtualization:` VITAL has developed mechanisms to support vitualized access to data generated by multiple IoT platforms and applications.
- `Modularity:` VITAL consists of a number of modules as depicted in Fig. 1. These components are developed as separate modules which can be deployed independently on different machines.
- `Standards-based:` VITAL uses a number of popular standards for data modelling e.g. RDF, JSON-LD, and ontologies. VITAL implements the Service Oriented Architecture (SOA) and enables RESTful web services for communication interchange mechanism.
- `Loosely Coupled:` VITAL in its nature is a service oriented distributed system which is developed around a loosely coupled approach.

- `Open Source:` VITAL is developed using open source technologies. To enable wide adaptation and maximize the impact, VITAL is distributed as an open source software under the LGPL license.

Figure 1 provides high-level architectural view of VITAL. Here we only provide an overview of VITAL and its components as the main focus of this article is the data models used in VITAL. At the lowest layer VITAL integrates multiple IoT systems and data sources (generated in multiple domains). The next layer is the Platform Provider Interfaces (PPI) that enables the access to metadata and data of IoT systems that are integrated. PPIs are responsible for mapping the data streams generated by IoT systems into VITAL data model for storage. The IoT Data Adaptor keeps track of registered PPIs, pulls data from IoT systems and stores it into the Data Management System (DMS). The DMS stores IoT data streams while enabling interfaces (for other components) to access stored (meta) data. VITAL platform also includes a number of added value functionalities such as Discovery Service for discovering IoT resources (e.g. sensors, systems, services, etc.), Filtering to filter information from DMS, Complex Event Processing (CEP) for event detection in IoT data streams, and Orchestrator for creating business specific services. Finally, the platform provides tools to allow the development of innovative cross-platform and cross-context applications and management and governance of IoT resources.

4 VITAL Data Model

Due to diverse set of use cases, VITAL covers a wide area of data models. It specifies the modelling of IoT systems and IoT services, e.g. using the Minimal Service Model (MSM) and basic IoT sensors and sensor measurements, e.g. using the Semantic Sensor Network (SSN) ontology. It also identifies data models for Smart City applications, e.g. smart transport systems. And it models the metadata for the VITAL system and its components, e.g. to model security and monitoring information. Next section provides an overview of the Linked Data technologies used in VITAL. Next we discuss existing ontologies and data models that are used as the basis of the VITAL data models and extended as needed by platform. This work is subdivided into different areas; sensors and sensor measurements, IoT systems and services, and Smart City Applications. We now list the ontologies we use for modelling the required data and also discuss the modelling of necessary additional data items.

4.1 Linked Data and Semantic

Linked Data (using RDF and ontologies) helps to describe and integrate data that is provided by different organisations in an interoperable way [37]. This is ideally suited for VITAL. VITAL envisages that a multitude of (independent) organisations and entities deploy and operate different sensors and IoT systems, which produce data (and offer services) that should be integrated in a system agnostic way. VITAL uses Linked Data standards for modelling and accessing metadata and data.

RDF: The Resource Description Framework (RDF) is World Wide Web Consortium (W3C)'s recommendations [22], which is also the most commonly used data model in the context of Linked Data. The RDF data model is a popular standard for describing things (known as resources or entities). By itself it is a graph-based data model that represents information as labelled directed graphs. This graph is built of triples that describe the data. Each triple (sub, pre, obj) consists of a subject sub, a predicate pre and an object obj. Using RDF in a Linked Data context has numerous advantages [37]:

- If the identifiers of data items (both used as subjects and objects) and vocabulary terms (used as predicates) are HTTP URIs, the RDF data model can be used at global scale and anybody is able to refer to anything.
- Each RDF triple is included in the Web of Data and can be the starting point for explorations in the data space, because any URI can be looked up in an RDF graph over the Web.
- It is possible to set RDF links between data from different sources.
- Sets of triples can be merged in a single graph to combine information.
- Terms taken from different vocabularies can be mixed in a RDF graph.

JSON-LD: A number of data formats are available that can be used to write RDF data, either directly as triples or as nodes that can be mapped to RDF triples, e.g. RDF/XML [22], RDFa [23], Turtle [30], N-Triples [21], and JSON-LD [12]. JSON-LD, a W3C recommendation, is a JSON-based serialization for Linked Data with the goals of simplicity, compatibility, terseness, expressiveness, etc. JSON-LD uses few important keywords, such as `@context, @id, @value,` and `@type`. In the VITAL platform JSON-LD is used. JSON-LD allows for referencing external files to provide context. This means contextual information can be requested on-demand which makes JSON-LD better suited to situations with high response times or low bandwidth usage requirements. Using JSON-LD will reduce the complexity of VITAL development by making it possible to reuse a large number of existing tools and reduce the inherent complexity of RDF documents.

Ontologies: Another important Linked Data concept used in VITAL is ontology. To integrate all data, generated by one or different sources, there have to be some rules. Some rules determine how the RDF graph is to be built and how triples may be connected or not. These rules are given by ontologies. An ontology specifies formally the conceptualisation of a domain of interest. As the conceptualisation is formal, a computer can automatically reason on it. There are many ontologies that have already been developed. Reuse of existing ontologies is crucial. If an existing ontology within the domain of use does not meet all the requirements and some new data models arise they should be attached to the existing ontology. We require ontologies in different areas. First, VITAL needs to model sensors and sensor measurements, which are the basis of any IoT system. Second, VITAL models IoT systems and services that are integrated into the VITAL. Third, VITAL provides means to model entities that are relevant to Smart City applications. And finally, it provides ontologies to model the VITAL system itself.

4.2 Sensors and Measurement

A number of ontologies have been developed to model sensors and sensor observations. Most of the existing ontologies are domain specific. Some abstract and generic ontologies are developed to provide a conceptual framework for IoT systems, for example, the Semantic Sensor Network (SSN) ontology developed by the W3C Semantic Sensor Network Incubator [35]. In practice such ontologies must be combined with additional ontologies to define concrete instances of abstract concepts. For instance, while a generic sensor ontology may specify how to model a sensor and its measurements, additional definitions must be used to model a concrete location of sensor and time when an observation was made. SSN is the generic sensor ontology that the VITAL consortium has selected to be used in VITAL.

SSN: The Semantic Sensor Network (SSN) ontology [35] defines a conceptual framework for describing sensors and sensor observations. The SSN ontology can formally describe sensors in terms of their:

- accuracy and capabilities,
- observations,
- measurement method,
- operating and survival ranges, and
- deployments.

A sensor could be any object that observes, it can be an electronic object, a virtual object or even a human. The ranges are used in the definition of sensors conjoined with the performance of these sensors. The description of deployment includes the deployment lifetime as well as the sensing purpose of the deployed macro instrument.

Time: Temporal aspects are essential for any system addressing real world phenomena, e.g. smart city IoT systems. Timestamps can be used to describe when a sensor observation was taken or when it was transferred. Multiple readings can be ordered by the time of their occurrence. Users may specify or query for certain types of observations based on specific timeframe. To model this, VITAL provides an ontology for time as well as temporal properties and relations. A well-established ontology for this is OWL Time [29]. OWL Time allows describing of temporal properties and relationships. It also supports time intervals as well as durations, which are useful for example, when describing imprecise measurement times as well as complex event specifications.

Location: Location in the physical world is another basic concept that is modelled in VITAL. There is a multitude of different location models and ontologies available today, including geographical and symbolic location models. VITAL follows a practical approach to allow easy usage of the system while at the same time being flexible enough for advanced use cases. WGS84 [31] coordinates are used as the basic location model, since they are the de-facto standard for outdoor localisation using the GPS system. In addition, symbolic names are often used as locations. VITAL uses the Linked GeoData system to model more complex location concepts, including symbolic names, cell-based locations and inter-location relationships.

Measurement: Different properties in the VITAL data models represent physical magnitudes like length or weight. Each one of these properties should be associated with an unambiguous unit of measurement, e.g. metre or kg. There is currently no single accepted ontology to model units of measurements in linked data. A number of potential ontologies were found and four were chosen for detailed evaluation. VITAL chooses QUDT [20] ontology for units of measurements due to the impressive scope and amount of information available on each type as well as the reputation of the maintainer and sponsoring party. QUDT is also actively maintained, with the latest version that was released in September 2016.

Modelling Sensor

To model sensors, sensor measurements and their descriptions in VITAL, we reuse and extend the SSN ontology and the Delivery Context (DC) [5] ontologies. A sensor is modelled as a `VitalSensor`, a subclass of `ssn:Sensor` and `dcn:Device`. By using the SSN ontology, VITAL can immediately describe sensors in detail, including aspects like the properties that they observe, sensor locations, and sensor observations. The SSN ontology also allows to model non-functional aspects of a sensor, e.g. its accuracy or reliability, by adding a `ssn:hasMeasurementProperty` property to the sensor description that points to a `ssn:MeasurementCapability`. The DC ontology defines a `dcn:Device` as a class that represents a device in the delivery context. By using the DC ontology, VITAL can reuse a highly detailed set of ontologies describing many aspects of devices, including their software, their hardware as well as their networking.

In addition to the SSN and DC ontologies, VITAL defines an additional property for sensors, `hasLastKnownLocation`. This property is a sub property of `dul:lastLocation` as specified in the SSN ontology description. It links to a location, which is the last known location of the sensor. The property does not imply that this is the actual current location of the sensor. If the sensor is mobile, it could have moved to a new location after the description was created. If the property is not available in a sensor description, then the location of the sensor may not be known.

Note that the location of a sensor can be modelled with different types as specified before, e.g. as a `geo:Point` in case GPS coordinates are used. To be as flexible as possible, we use the generic `dul:Entity` class to represent all different location types here. It is taken from the DOLCE+DnS Ultralite (DUL) ontology and defines it as anything: real, possible, or imaginary, which some modeller wants to talk about for some purpose [17]. In addition to the basic description of a sensor, some VITAL services may require additional information. Properties and classes to model this information are described in next sections.

Modelling Sensor Measurements

Similarly to sensors, VITAL uses the SSN ontology to model sensor measurements. A measurement is modelled as an `ssn:Observation`. The observation contains a link to an observed property (using `ssn:observedProperty`) to specify what the observation is measuring. In addition, it specifies when the measurement was taken (`ssn:observationResultTime`), at which location (`dul:hasLocation`) in WGS84 format, the quality of the measurement (`ssn:observationQuality`), as well as the measured value (`ssn:observationResult`) with the unit of measurement specified with the QUDT ontology.

4.3 IoT Systems and Services

VITAL integrates existing IoT systems (e.g. deployed platforms) and allows clients (applications as well as VITAL system services) to access (meta) data and services of such systems. Currently there are four platforms for which example deployments are integrated as a proof of concept: X-GSN, Hi-Reply, INRIA FIT, and Xively. To integrate systems and work with them, VITAL needs a set of models to describe IoT systems and their services.

Modelling Systems

VITAL models an `IotSystem` as a subclass of `ssn:System` with a number of additional properties. An IoT system description always includes a basic set of properties that describe general aspects of the system, e.g. its operator. In addition, a system description may specify a set of IoT services that it offers. To describe general metadata about the system, VITAL supports three new properties: `status`, `operator`, and `serviceArea`. The status of an IoT system might change during its lifecycle, thus, VITAL compliant systems that want to expose their current operational stat must manage a virtual sensor of type `MonitoringSensor` (a sub class of `VitalSensor`). Furthermore, `OperationalState` specifies the operational state of a system. A number of states are defined in VITAL as sub classes of `OperationalState`: `Operational`, `StartingUp`, `Running`, `ShuttingDown`, and `Unavailable`. In addition to the metadata discussed so far, an IoT system may offer a set of IoT services to access its functionalities. To allow an IoT system to link to descriptions of provided IoT services, VITAL introduces a new property `providesService`.

Modelling Services

In VITAL, an IoT system does not only provide access to IoT data (e.g. sensor measurements) but may offer a set of distinct and heterogeneous IoT services. An IoT service may be generic, e.g. a service to discovery ICOs or to access filtered data, or application specific, e.g. a service to reserve a parking space in a Smart City IoT system. In fact, VITAL models all functionality that can be exposed by an IoT system and can be accessed and used by a client as an IoT service, including data access, e.g. reading a sensor measurement. VITAL therefore specifies a flexible data model to specify all different kinds of IoT services. There is currently no single, standardised way to model IoT services. Based on the related work discussed before, we decided to base VITAL's semantic IoT service model on existing work in the domain of web services.

As discussed before, VITAL aims at providing a semantic model that is generic – yet simple and minimal, reuses existing ontologies as much as possible and allows to link with an active community as well as other current projects. After careful consideration, we selected to use widely used Minimal Service Model (MSM) [13] as the basis of its IoT service modelling ontology. In the VITAL system an IoT service is modelled as a RESTful (web) service that is described by Linked Data using the MSM ontologies. This allows publishing a description of the IoT service that can e.g. be used for discovery or for automatic composition tasks.

IoT systems (integrated within VITAL platform) may provide configuration functionalities. In order to model these functionalities, `ConfigurationService` class is defined as a sub class of `msm:Service` along with two operations: `GetConfiguration` in order to access existing configurations and `SetConfiguration` to set new configurations.

An IoT system can allow VITAL to monitor a number of monitoring functionalities. For example, the status of an IoT system, the status of sensors that an IoT system manages, performance metrics of an IoT system, SLA parameters related to an IoT system, etc. These functionalities are exposed by a `MonitoringService` class a sub class of `msm:Service` with a number of operations: `GetSystemStatus`, `GetSensorStatus`, `GetSupportedPerformanceMetrics`, `GetPerformanceMetrics`, `GetSupportedSLAParameters`, and `GetSLAParameters`.

The VITAL platform can use both a pull and push based mechanism to obtain observations made by a sensor. An IoT system with various sensors can provide/support both mechanism by providing an observation service. An IoT system must support at least one of these two mechanisms in order to allow access to sensor observations. This service is modelled as `ObservationService` sub class of `msm:Service` with the following operations: `GetObservations`, `SubscribeToObservationStream`, and `UnsubscribeFromObservationStream`.

4.4 Smart Cities

The VITAL platform is under proof-of-concept validation using the two use cases focus on Smart Transport and Traffic Management and Smart Working. Therefore, in the following we discuss how to model data items and properties that are relevant for these two scenarios. Clearly, VITAL is not restricted to these two use case scenarios. A user who would like to use VITAL for other smart city aspects can do so by specifying additional ontology elements. Due to the nature of Linked Data, these additional elements can be added at any time without the need to redesign the system.

Modelling Cities

VITAL obtains the majority of its semantic information on cities from DBpedia, using the classic DBpedia dataset for most information with the option of using DBpedia live for information that updates more frequently. It is also encouraged to link real places and services in cities back to DBpedia to improve the amount of knowledge available. For example, while Camden Road would be modelled as an `otn:Road` as part of a smart transport system, it should also link to `dbpedia:Camden_Road`. As discussed, `ssn:observes` is used to specify a property of the real world a sensor is observing. The specification for modelling real instances of such properties are application domain specific and should be defined as required. For these two use cases we re-use the ontology of FP7 project OpenIoT [40].

Modelling Smart Transport

VITAL models transport infrastructure using a combination of ontologies. The core of these is the Ontology of Transportation Networks (OTN) [16]. This ontology allows easy modelling of a transport network graph with connections between infrastructure such as bus and train stations as well as events such as accidents and blocked passages. In order to model the traffic management system, VITAL describes a class `TrafficManagement` a sub class of `IoTSystem`. For the purpose of transport and traffic scenarios and use cases (specifically in Camden Town, London), VITAL models the following sub classes of `ssn:Property`: `BusArrival`, `RailArrival`, `TubeArrival`, and `AvailableBikes`. To describe general description of `Line`, VITAL supports two new properties. These properties are: `name` and `direction`

Modelling Smart Working

The structural development in advanced economies is influencing the change in working patterns with (increasingly) employees more likely to work on the move. Mobile workers require optimal working environments to be available at short notice and without any difficulties to use. Owners of these suitable environments require optimal occupancy. To meet these requirements, the smarter working application (powered by VITAL) introduced a number of sub classes of ssn:Property: AvailableDesks and Availability. Each available workspace in the system that meets the specified criteria is shown on a map and/or list. Additional information is also shown associated with each workspace e.g. anticipated air quality, temperature, humidity, and footfall in the requested time window and location etc. In order to model additional information, following classes are defined as sub classes of ssn:Property: CarbonMonoxide, Ozone, and Footfall. Should the additional information need to be added and modelled by introducing new classes, they can be added as required.

4.5 JSON-LD Definitions

Since Linked Data in VITAL is always formatted as JSON-LD we introduce some additional definitions (in a JSON-LD context section) that do not change the used ontology or the resulting RDF triples but align the JSON-LD representation more closely to normal JSON and thus makes it easier for developers to work with the data. All JSON keys that do not specify a prefix will be expanded to URIs in the VITAL ontology namespace. This results in more compact files with less clutter. Then we define that the key id will be mapped to a JSON-LD node identifier (@id). The node identifier is used to create the URI that is used as the subject in RDF triples. Similarly, specified key will be mapped to the JSON-LD keyword @type. This results in an RDF triple being created that specifies the RDF type of a node. To further simplify the JSON-LD file, the key name will be mapped to rdfs:label and the key description will be mapped to rdfs:comment. We then specify a number of prefixes that can be used in the JSON-LD description to reduce the length of keys by specifying them as so-called terms. All these mappings are completely transparent to developers and can be ignored by clients. They are only relevant if the JSON-LD file is mapped to RDF triples internally. Together, they reduce the complexity of the resulting JSON-LD file and make it both smaller and easier to read and understand for JSON developers. The resulted JSON-LD contexts are available at:

- http://vital-iot.eu/contexts/system.jsonld
- http://vital-iot.eu/contexts/service.jsonld
- http://vital-iot.eu/contexts/sensor.jsonld
- http://vital-iot.eu/contexts/measurement.jsonld

4.6 VITAL Ontology

In previous sections we discussed the VITAL data models e.g., the major classes, properties, and operations that are used to model IoT systems, services, sensors, observations, and smart city applications. Clearly, discussion on each concept is impossible. Figure 2 visualizes the main entities in VITAL ontology. VITAL ontology can be accessed at: http://vital-iot.eu/ontology/ns/ontology.owl.

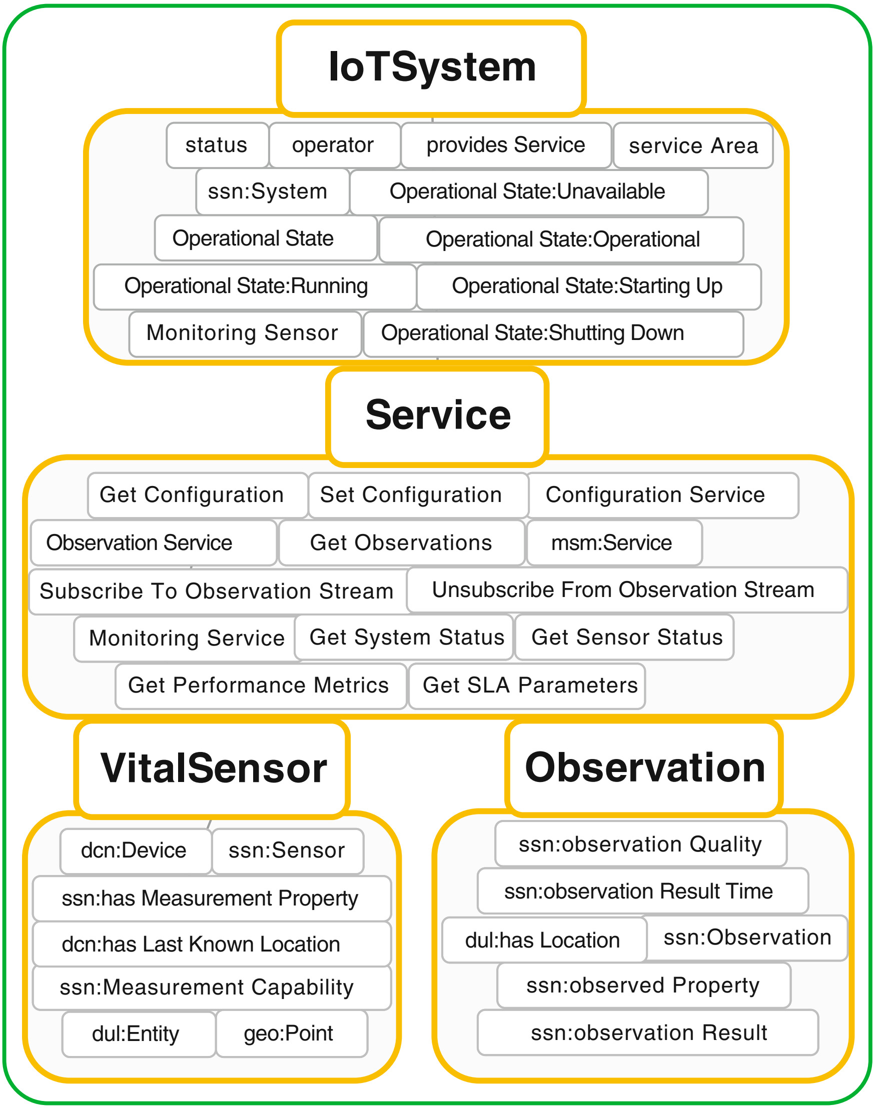

Fig. 2. VITAL data model: entities and their modelling

5 Conclusion and Future Work

The Internet of Things (IoT) domain has gained a significant attention from both academia and industry. This has led to the development of multiple heterogeneous architectures and standards, platforms, and IoT applications. Clearly, there is a pressing need that IoT applications address the challenge of heterogeneity and allow the exchange of information across platforms and applications.

This paper provides the basis for the semantic (meta) data models used in the VITAL system. We build upon Linked Data principles and technologies to provide interoperable and platform agnostic data models that are based on existing ontologies. This allows VITAL applications to integrate other data sources in the Web, resulting in a large and varied set of usable data items. Although we analysed a large number of ontologies during the design of the VITAL data models, the work is not finished. Thus, it is envisaged that more data models will be added as VITAL extends its functionalities and more VITAL-powered use cases are developed.

Acknowledgment. The authors acknowledge support from the ICT-2013.1.14 - VITAL-IoT project, which is co-funded by the European Commission under the Seventh Framework Program (FP7), grant agreement number 608682. The authors also acknowledge help and contributions from all partners of the VITAL consortium.

References

1. The alliance for internet of things innovation (aioti). http://www.aioti.eu. Accessed 4 Sep 2016
2. The architectural reference model (arm) - iot-a. http://www.iot-a.eu. Accessed 14 Sep 2016
3. Bridging the interoperability gap of the internet of things (big iot). http://big-iot.eu/. Accessed 10 Oct 2016
4. Connecting the real world: Parklight. Technical Report. Accessed 4 Sep 2016
5. Delivery context ontology. https://www.w3.org/TR/2009/WD-dcontology-20090616. Accessed 14 Sep 2016
6. The electronic product code (epcglobal). http://www.gs1.org/epcglobal. Accessed 4 Sep 2016
7. The eu-funded project burba. http://www.burbaproject.net. Accessed 4 Sep 2016
8. The european research cluster on the internet of things (ierc). http://www.internet-of-things-research.eu. Accessed 4 Sep 2016
9. Fit. https://www.fit-equipex.fr. Accessed 4 Sep 2016
10. Gea farm technologies: Heatwatch. http://www.jacesupplies.co.uk. Accessed 4 Sep 2016
11. Hireply. http://www.reply.eu. Accessed 4 Sep 2016
12. Json-ld 1.0: A json-based serialization for linked data. https://www.w3.org/TR/json-ld. Accessed 14 Sep 2016
13. Minimal service model. http://iserve.kmi.open.ac.uk/ns/msm/msm-2013-05-03.html. Accessed 14 Sep 2016
14. Netatmo. https://www.netatmo.com. Accessed 4 Sep 2016
15. The onfarm system: grow informed. http://www.onfarm.com. Accessed 4 Sep 2016
16. Ontology of transportation networks: Deliverable a1–d4, project rewerse: Reasoning on the web with rules and semantics. http://rewerse.net/deliverables/m18/a1-d4.pdf. Accessed 14 Sep 2016
17. Ontology:dolce+dns ultralite ontology. http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite. Accessed 14 Sep 2016
18. The open geospatial consortium (ogc). http://www.opengeospatial.org. Accessed 4 Sep 2016

19. Open source implementation of onem2m. http://www.eclipse.org/om2m. Accessed 11 Nov 2016
20. Qudt - quantities, units, dimensions and data types ontologies. http://qudt.org. Accessed 14 Sep 2016
21. Rdf 1.1 n-triples: A line-based syntax for an rdf graph. https://www.w3.org/TR/n-triples. Accessed 14 Sep 2016
22. Rdf 1.1 xml syntax. https://www.w3.org/TR/rdf-syntax-grammar. Accessed 14 Sep 2016
23. Rdfa 1.1 primer - third edition: Rich structured data markup for web documents. https://www.w3.org/TR/rdfa-primer. Accessed 14 Sep 2016
24. Smart street lighting. http://www.echelon.com. Accessed 4 Sep 2016
25. Smart structures: Smartpile. http://smart-structures.com. Accessed 4 Sep 2016
26. Standards for m2m and the internet of things. http://www.onem2m.org. Accessed 11 Nov 2016
27. Thingsspeak. https://www.thingspeak.com. Accessed 4 Sep 2016
28. Thingworx an iot platform. http://www.thingworx.com. Accessed 4 Sep 2016
29. Time ontology in owl. https://www.w3.org/TR/owl-time. Accessed 14 Sep 2016
30. Turtle - terse rdf triple language. https://www.w3.org/TeamSubmission/turtle. Accessed 14 Sep 2016
31. W3c geo positioning. https://www.w3.org/2003/01/geo/wgs84_pos. Accessed 14 Sep 2016
32. The w3c semantic sensor network (ssn) incubator group. https://www.w3.org/2005/Incubator/ssn. Accessed 4 Sep 2016
33. Xively. http://xively.com. Accessed 4 Sep 2016
34. International Electrotechnical Commission: Orchestrating infrastructure for sustainable Smart Cities (2014)
35. Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W. D., Phuoc, D. L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K.: The {SSN} ontology of the {W3C} semantic sensor network incubator group. Web Semant. Sci. Serv. Agents World Wide Web **17**, 25–32 (2012)
36. Dixon, C., Mahajan, R., Agarwal, S., Brush, A., Lee, B., Saroiu, S., Bahl, P.: An operating system for the home. In: The 9th USENIX Symposium on Networked Systems Design and Implementation (2012)
37. Heath, T., Bizer, C.: Linked data: evolving the web into a global data space (2011)
38. Kazmi, A.H., O'Grady, M.J., Delaney, D.T., Ruzzelli, A.G., O'Hare, G.M.P.: A review of wireless-sensor-network-enabled building energy management systems. ACM Trans. Sen. Netw. **10**(4), 66: 1–66: 43 (2014)
39. Kazmi, A.H., O'Grady, M.J., O'Hare, G.M.P.: Energy management in the smart home. In: Proceedings of the 2013 IEEE 10th International Conference on Ubiquitous Intelligence & Computing and 2013 IEEE 10th International Conference on Autonomic & Trusted Computing (UIC-ATC 2013), pp. 480–486. IEEE Computer Society, Washington, DC (2013)
40. Serrano, M., Kefalakis, N., Soldatos, J., Hauswirth, M.: OpenIoT: an open source platform for enabling intelligence to the internet of things (2014)
41. Smith, I.G., Vermesan, O., Friess, P., Furness, A.: The Internet of Things 2012 New Horizons (2012)
42. Vermesan, O., Friess, P.: Internet of things - from research and innovation to market deployment (2014)

Springer

http://www.springer.com/978-3-319-56876-8

Interoperability and Open-Source Solutions for the
Internet of Things
Second International Workshop, InterOSS-IoT 2016,
Held in Conjunction with IoT 2016, Stuttgart, Germany,
November 7, 2016, Invited Papers
Podnar Žarko, I.; Broering, A.; Soursos, S.; Serrano, M.
(Eds.)

2017, X, 187 p. 50 illus., Softcover
ISBN: 978-3-319-56876-8