International Journal of Advances in Computer Science & Its Applications
Volume 6 : Issue 1 [ISSN 2250-3765]
Publication Date : 18 April, 2016

A Survey on Data Models for Internet of Things

Wenbin LI

Abstract— **In Internet of Things (IoT), heterogeneous objects are connected as networks to facilitate the exchange of goods and services. These objects, varying from electronic devices to physical entities, rely on data models to be accessed and achieve their digital functionality. Nevertheless, we observe a divergence of data models in current IoT environments which has been influencing data exchange, share and integration from different sources. In this paper, we present a survey on IoT data models and provide background for a common understanding. Firstly, we introduce the concept of IoT and its emergence; and then we discuss IoT data models according to data structure and management operations to show current achievements and limitations; at last we illustrate the challenges and prospects to shed light on future work.**

Keywords— **Internet of Things, Data Model, Survey, Semantics**

I. Introduction

In modern society, we live surrounded by a growing number of smart objects to improve life efficiency, accuracy and economic benefits in an intuitive way. Smart objects are autonomous physical/digital objects augmented with sensing, processing, and network capabilities [1], which does not only include electronic devices such as mobiles and sensors, but also legacy physical entities such as cars and rooms, embedded with electronic tags such as Radio-Frequency Identification (RFID) tags [2] or sensors. Industry leaders predict that the number of connected devices will surpass 15 billion nodes by 2015 and reach over 50 billion by 2020 [3], not to mention the total number of smart objects. Such a large number of connected objects highly rely on the reactivity, autonomy, scalability and dependability of their underlying infrastructure, while the enormous volume of data produced every day greatly increases the complexity of design and maintenance of its enabled systems based on Information and Communication Technologies (ICT).

Under these circumstances, Internet of Things (IoT) emerges as the ICT revolution that represents the future combination of computing and communications. IoT is a network of physical objects that can be discovered, monitored, controlled or interacted with by electronic devices which communicate over various networking interfaces, and eventually can be connected to the wider Internet [4]. In this network, objects are sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems [5].

In order to store objects and process computing in IoT, the objects are typically represented as data model. Data model defines how data are presented, how they are connected to each other and how they are stored and processed. The combination of the physical objects with their data models constitutes the basic unit in IoT. As the "Internet representation" of physical objects in IoT, data model is attracting more and more research attention to ensure data manipulation and process in networks. A variety of IoT data models have been proposed such as declarative model NGSI-10 [6], programming model OSGi DAL [7], ontology model Semantic Sensor Network (SSN) Ontology [8], etc., and each of them has different data structures and characteristics. In this paper, we try to provide a global view of IoT data models by presenting and comparing representative propositions from different perspectives. The objective is to define a framework for a common and understandable view to guide future research initiatives and contributions. The remainder of the paper is organized as follows: firstly, we present a brief introduction for IoT and its emergence; secondly, we analyze a list of IoT data models and compare their advantages; at last, we conclude several challenges in current research and shed light on future work for IoT data models.

II. IoT at a Glance

IoT combines computing and communicating to make benefits from the collective effect of object networks, and such a huge collection requires the support of a number of technologies to ensure the functionality and performance of networks and systems. The emergence of IoT is due to the significant advance and convergence of a number of technologies, which are surveyed in [5][9][10]. These technologies vary from Nanoelectronics, Identification and Tracking Technologies, Sensor Networks and Embedded Systems to Machine-to-Machine, Cloud Computing, Distributed Intelligences, Big Data and Semantic Data, just to mention a few. In our work, we group the technologies into three categories that led to the emergence of this new research field.

A. *Development of Manufacturing and Embedding Technology*

The desire to have more functionality in increasingly smaller size has driven the development of manufacturing to delivery smaller and smarter sensors, actuators and other computing chips, while embedding technology enables the integration of different devices into complex devices to provide more mechanical and electrical functions, and also makes it possible to transform legacy physical entities into smart objects that can understand and react to IoT environments. With the development of manufacturing and embedding technology, heterogeneous objects in IoT are

Wenbin LI
Orange Labs,
Grenoble, France

instrumented and we now have the ability to measure, sense and control smarter objects in real-time environment.

B. Advance in ICT

ICT consist of all technologies used to handle information and enable communication among computers and network hardware, communication middleware and software [11]. The advance in ICT has greatly facilitated information exchanges regarding resources, applications, services, knowledge, and driven the development of complex and distributed systems. IoT users, objects, and systems are interconnected and enabled to effectively communicate with each other.

C. Evolution of Computational Intelligence

Now information and communication systems are invisibly embedded in IoT environment, while Computational Intelligence provides a set of computational approaches to process vast set of data generated from diverse objects in diverse locations. More importantly, the evolution of Computational Intelligence presents methodologies for IoT to respond to the increasing system demands and environmental changes more quickly, accurately and intelligently.

With the support of aforementioned technologies, heterogeneous objects in IoT appear as smart objects with sensing, communicating and computing capabilities. Each object is represented by its data model and is able to interoperate within the existing Internet infrastructure.

III. IoT Data Models

Drawing upon generic Information Technology (IT) data models, IoT data models describe the physical nature of things in addition to their digital nature. In the following, we introduce a representative set of IoT data models and discuss their characteristics.

A. NGSI-10

NGSI-10 [6] is a purely declarative model that describes both electronic devices and physical objects. The central aspect of the NGSI-10 is the concept of *entities*, which are the virtual representation of all kinds of physical objects in the real world, such as person, place, event, etc., and any available information about entities is expressed in the form of *attributes* which are composed of attribute name and attribute type. In addition to this, a context management component provides the NGSI-10 with RESTful interfaces to manage context information about entities with operations as follows [12]:

- Update: provide context information;
- Query and Subscribe/Notify: consume context information;
- Register and Discover: discover entities through query or notifications.

NGSI-10 API only supports XML as data serialization format.

B. DPWS

Devices Profile for Web Services (DPWS) defines a minimal set of implementation constraints to enable secure Web Service messaging, discovery, description, and eventing on resource-constrained devices [13]. Aligning with Web Service Definition Language (WSDL) [14], DPWS additionally introduces the concept of hosting to define the relationship between a hosted service and its hosting device, and the concept of characteristics to present manufacturer information.

DPWS focuses on allowing for seamless integration of device-provided by providing the following functionality [15]:

- Discovering DPWS-capable devices on the network and the services they offer
- Sending messages to DPWS-capable devices and receiving replies
- Describing a Web service by providing a WSDL file
- Interacting with a service using its description
- Subscribing to and receiving events from a Web service

DPWS provides data model only for devices and since DPWS is aligned with Web service technology, it only supports pure functional abstraction of devices but has no uniform interface for device management functions.

C. SensorML

Sensor Model Language (SensorML) provides standard models and an XML encoding for describing sensors and measurement processes [16]. Different from previous models, the focus of SensorML is to define processes and processing components associated with the measurement and post-measurement transformation of observations, which includes sensors ("things that measure"), actuators ("things that act"), and processors ("things that calculate").

In SensorML, all components are modeled as processes. This includes physical components such as detectors, actuators, and physical processors, sensors and platforms (which are viewed as physical systems). All components are modeled as processes that receive input and through the application of an algorithm defined by a method and set parameter values, generate output. The supported operations include: sensor discovery, sensor geolocation, processing of sensor observations, and subscription to sensor alerts. However, this model highly relies on traditional SOA architecture, and requires users' familiarity to interact with different components.

D. UPnP

UPnP is a set of networking protocols that permits networked devices, such as personal computers, printers, Internet gateways, Wi-Fi access points and mobile devices to seamlessly discover each other's presence on the network and establish functional network services for data sharing, communications, and entertainment [17]. Among UPnP technologies, data model in UPnP is similar to DPWS which is expressed in XML and includes vendor-specific manufacturer information like the model name and number, serial number, manufacturer name, URLs to vendor-specific web sites, etc. UPnP data models are stored and interacted with via services of sensor management database, which contains methods to define, create, delete data records, and to select, identify and retrieve data record contents.

Since previous version UPnP 1.0 supports only devices based on UPnP technologies, the objective of UPnP 2.0 is to host services to bridge sensor devices connected to both UPnP networks as well as non-UPnP based networks. UPnP 2.0 allows new IoT devices to be added without creating whole new specifications, and existing APIs are being mapped to REST+JSON to provide pure RESTful interfaces.

However, current UPnP data model has no explicit notions of state to model device from system-theoretic sense, and UPnP provides no support for non-devices entities.

E. LWM2M

Lightweight Machine To Machine (LWM2M) [18] specifies a set of common interfaces and data models to enable plug and play interoperability between CoAP devices and local or remote services. LWM2M defines a simple data model where each atomic piece of information is a *Resource*. A resource can be written, read and executed. Resources are logically organized into *Objects*, while an object is a collection of resources. Six normative objects are defined in LWM2M as LWM2M server, Access Control, Device, Connectivity Monitoring, Firmware, and Location.

In addition, LWM2M provides four interfaces as follows to support server and client interaction:

- Bootstrap: Pre-provisioned or Client/Server Initiated
- Client Registration: Register the Client and its Objects
- Device Management and Service Enablement: Server access to Object or Resource
- Information Reporting: Notifications with new resource values

LWM2M supports both syntactic and semantic description of data model and message exchange.

F. OSGi DAL

OGSi Device Abstraction Layer (DAL) is a programming data model to achieve an abstraction layer that unifies the work with devices supporting different protocols (e.g., ZigBee, Z-Wave, KNX, and UPnP) [7]. The application programmers can work with devices provided by different protocols exactly in the same way and by applying the same program interface. Generally in OSGi DAL, a device is described by a set of properties and has functions as basic management operations; a function provides function data, and posts function events for property changes.

OSGi DAL improves interoperability of the applications and the devices under heterogeneous environment including devices that support different protocols. As a low-level programming model, the applications can work without modification when new hardware controllers and protocol adapters are dynamically added.

G. openHAB

The open Home Automation Bus (openHAB) is another programming model containing a set of OSGi bundles deployed on an OSGi framework for integrating different home automation systems and technologies into one single solution that allows over-arching automation rules and that offers uniform user interfaces [19].

The core concept in openHAB is the notion of item. An item is an abstract object linked to a bundle which can be a particular device in the physical world or some value of it. All features offered by openHAB use “item” abstraction, and no difference between device or device reference (e.g., IP and id), which makes it easy to replace one technology by another. openHAB also use action to plug additional functionality in the openHAB system that can be manipulated via scripts. Each action is typically a separate Java class that provides simple methods that provide the particular functionality.

Similar to OSGi DAL, openHAB is based on OSGi, and it provides highly modular architecture which allows adding and removing functionality during runtime without stopping the service. Since openHAB uses item as the same abstraction for devices, device properties and device references, it is not expressive enough to concretize different devices with different variants.

H. Ontologies

The IoT support for abstraction and semantics is being pushed forward by several standardization organizations such as the ETSI M2M in [20], OneM2M in [21] and W3C Web of Things [22]. Ontology provides a formal, explicit specification of a shared conceptualization of a domain [23]. In the context of IoT, ontologies are expected to define resources (e.g. sensors), observation and measurement data (e.g. temperature), domain concepts (e.g. location), services (e.g. device functions) and other data sources. By use of ontologies, semantics are added to data models to create common understanding of data among people and system, and also facilitate data sharing and reuse from different sources. In this paper, we introduce three IoT ontologies, i.e., OneM2M Ontolgoy, SSN and IoT-A.

1) OneM2M Base Ontology

OneM2M is a global initiative to ensure the most efficient deployment of Machine-to-Machine (M2M) communications systems and IoT, while OneM2M base ontology constitutes a basis framework for specifying the semantics of data that are handled in OneM2M [24]. In OneM2M base Ontology, a Thing in oneM2M is an entity that can be identified in the oneM2M system, while a ThingProperty denotes an attribute of a thing that can be can be described by data. A Device is an object designed to accomplish a particular task. In order to accomplish its task, the device performs one or more Functionality, and must offer at least one Service. OneM2M ontology provides a basic ontology reference to model devices, services, functions, and operations in IoT. As an ontology being developed, the base ontology replies on interworking with other IoT ontologies to model IoT devices.

2) SSN

SSN ontology describes the capabilities and properties of sensors, the act of sensing and the resulting observations [7]. The full ontology consists of 41 concepts and 39 object properties, and the main classes are device, observation, feature of interest, sensing process, deployment, platform and measurement capability. The ontology can be broadly divided into four clusters: a sensor cluster with a focus on what senses, how it senses, and what is sensed; an observation cluster with a focus on observation data and related metadata; a system cluster, with a focus on systems of sensors and deployments; a feature and property cluster, focusing on what senses a particular property or what observations have been made about a property.

SNN presents a detailed vocabulary without providing specific ontology for real devices. In SSN, sensor related but not specific materials such as units of measurement, locations, hierarchies of sensor types, and property hierarchies are not defined. As ontology for sensor network, SSN lacks classes and properties related to sensor communication process.

3) IoT-A

As part of the FP7 IoT-Architecture project, IoT-A ontology defines a set of vocabularies for entities, resources, devices and services by extending SSN [25].

In IoT-A, a resource is the core software component that represents an entity in the digital world. It allows the entity to be part of the digital world by mediating the interactions; the entity model defines required properties of an identifier and attributes; resources are accessed by services which provide functionality to gather information about entities they are associated with or manipulate physical properties of their associated entities.

Comparing to SSN, IoT-A ontology introduces the entity model to support IoT entities in addition to devices, and includes measurement units and general domain knowledge that need to be associated with the sensor data. IoT-A ontologies still does not provide any ontology for specific devices or ontologies but rather require users to construct their own model by use of vocabulary structure.

Ontology is currently the most popular model to create semantic data and promote interoperability among different data sources. Apparently, the creation of data model based on ontologies requires extra development efforts; to fully reuse existing syntactic data, serialization formats are needed to support semantic annotation to existing data. Moreover, in most cases users have to combine a set of domain-independent and domain-specific ontologies to model their own data since no ontology exists covering all related information in all domains.

4) LOV and LOV4IoT

The Linked Open Vocabularies (LOV) [26] is a catalogue, created by the semantic web community which references 527 well - designed vocabularies, i.e., ontologies, according to the semantic web best practices. A vocabulary in LOV gathers definitions of a set of classes and properties (aka. terms of the vocabulary), to describe specific types of things, things in a given domain or industry, things at large but for a specific usage, as well as links between things. The definitions of terms provided by the vocabularies bring clear semantics to descriptions and links.

Linked Open Vocabularies for the Internet of Things (LOV4IoT) [27] references more than 290 domain ontologies which cannot be referenced on the LOV catalogue since they do not respect the semantic web best practices. The ontologies are classified by domains such as building automation, healthcare, security, weather forecasting, intelligent transportation systems, affective science, tourism, agriculture, food, etc.

LOV and LOV4IoT gather good ontology references to category domain semantic information and reuse domain knowledge expertise, and the number of ontologies within keeps growing to provide richer semantic model for IoT.

I. *Comparison*

We hereby present a brief comparison of all abovementioned IoT data models according three criteria as follows:

- **Abstraction Level**: A data model is *device-centric* if it defines data model only for electronic devices; a data model is *entity-centric* if it defines data model for both electronic devices and physical entities.
- **Interface**: Since some of data models provide functionality to support data operations (e.g., discover, update, query, etc.), we compare interfaces that data models present for data operations. The interfaces include Web interface based on *SOAP/REST/CoAP*, *programming API*, and the interfaces for ontologies are *not specified* as they depend on their applications.
- **Semantics**: A data model is *semantic* if it includes semantic information defining the meaning of its instance; otherwise, it is *syntactic*.

Table 1 illustrates the comparison of IoT data models.

TABLE I. COMPARISON OF IOT DATA MODELS

	Abstraction Level	**Interface**	**Semantics**
NGSI-10	Entity-Centric	REST	Syntactic
DPWS	Device-Centric	SOAP	Syntactic
SensorML	Device-Centric	SOAP	Syntactic
UPnP	Device-Centric	SOAP and REST	Semantic*
LWM2M	Device-Centric	CoAP	Semantic*
OSGi DAL	Device-Centric	Programing API	Syntactic
OpennHAB	Device-Centric	Programing API	Syntactic
oneM2M	Device-Centric	Not Specified	Semantic
SSN	Device-Centric	Not Specified	Semantic
IoT-A	Entity-Centric	Not Specified	Semantic
LOV **LOV4IoT**	Entity-Centric	Not Specified	Semantic

*: Syntactic model with support Semantic annotation

From Table 1 we can see that few data models (i.e., NGSI-10 and IoT-A) are able to define both IoT devices and entities; Web interface is more friendly for users to manipulate data, and UPnP newly presents REST interfaces in addition to its typical SOAP interface; besides ontologies, UPnP and LWM2M supports explicitly semantic annotation to existing data, while other data models only present syntactic data.

IV. Challenges

Following the comparison of IoT data models, we identify fours challenges to drive future research on IoT data model.

- **Entity-centric data model**: IoT objects do not only include electronic devices but also physical entities. Obviously data model for both device and entity is obligatory for future IoT research, and this requires data models to take into account of heterogeneity of IoT entities, as well as their physical properties in addition to their digital nature.
- **Semantic-supported data model**: Without semantic information in data model, most applications are vertically integrated, and many legacy systems (e.g. security) are non-connected or closed. Semantic-supported data model is expected to interconnect different devices and entities, and the semantic-supported data model can either be data model based on ontologies or existing data models extensible with semantic annotations.
- **Rich data model**: Lack of rich data models results in extra modelling effort of searching through different data models (e.g., ontologies). In addition, shared environment models for applications that share same environment is missing (e.g., smart homes and smart cities), and thus exploitation of leveraging invariants from one environment instance to another is difficult.
- **Automatic semantic annotation**: Creating ontologies and defining rich data models are not enough, as data model often requires manual association with ontologies. Automatic solutions to create semantic data, semantically annotate data and link with other data sources are expected to reduce human intervention and promote the popularity of semantic Web.

V. Conclusion

IoT is an emerging global Internet-based information architecture that allows objects to be sensed and controlled remotely across existing network infrastructure. IoT data models organize data elements of IoT objects and standardize how data are processed and connected. This paper presents a picture of the current state of the art on IoT data models. More specifically, it firstly introduces the concept of IoT and its emergence; and then it reviews a representative set of IoT data models from different perspectives, i.e., declarative model, programming model and ontology model; at last presents an analysis of the major research issues which remains to be solved.

References

[1] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, “Smart Objects as Building Blocks for the Internet of Things,” Internet Computing, IEEE, vol. 14, no. 1, pp. 44–51, 2010.

[2] Shepard, Steven. RFID: radio frequency identification. McGraw Hill Professional, 2005.

[3] Silicon Labs, Bringing the Internet of Things to Life, White Paper, 2012.

[4] D. D. Guinard and V. M. Trifa, Building the Web of Things Book | Web of Things. Manning Publications, 2015.

[5] O. Vermesan and P. Friess, Internet of things: converging technologies for smart environments and integrated ecosystems. 2013.

[6] Open Mobile Alliance, NGSI Context Management, 2010

[7] OSGi Alliance, Device Abstraction Layer, 2014

[8] M. Compton, P. Barnaghi, L. Bermudez, R. G. Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, and A. Herzog, “The SSN Ontology of the Semantic Sensor Networks Incubator Group,” Web Semant, vol. 17, pp. 25–32, 2011.

[9] L. Atzoria et al., “The Internet of Things: A survey,” Computer Networks, vol. 54, no. 15, 2010, pp. 2787-2805.

[10] A. Whitmore, A. Agarwal, and L. D. Xu, “The Internet of Things—A survey of topics and trends,” Inf Syst Front, vol. 17, no. 2, pp. 261–274, Mar. 2014

[11] Elisha, M.J. The application of Information and Communication Technology in Nigerian academic libraries prospects and problems, J. Information Manager 6, 1 (2008)

[12] FI-WARE NGSI-10 Open RESTful API Specification,
https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_NGSI-10_Open_RESTful_API_Specification

[13] S. Chan et al. “Devices Profile for Web Services,” 2006

[14] “Web Service Definition Language (WSDL).” [Online]. Available: http://www.w3.org/TR/wsdl. [Accessed: 01-Nov-2015].

[15] Introducing DPWS, https://msdn.microsoft.com/en-us/library/dd170125.aspx

[16] Open Geospatial Consortium, SensorML: Model and XML Encoding Standard, 2014

[17] B. Middleton, Conducting Network Penetration and Espionage in a Global Environment. CRC Press, 2014.

[18] Open Mobile Alliance, Lightweight Machine to Machine Technical Specification, 2015

[19] openHAB - Features – Introduction, http://www.openhab.org/features/introduction.html

[20] ETSI, Study on Semantic support for M2M data, 2013

[21] OneM2MTechical Report, Study of Abstraction and Semantics Enablements, 2015

[22] Web of Things at W3C, http://www.w3.org/WoT/

[23] M. M. Taye, “Understanding Semantic Web and Ontologies: Theory and Applications,” arXiv preprint arXiv:1006.4567, 2010.

[24] oneM2M Technical Specification, TS-0012-V-0.2.0, 2015

[25] S. De, T. Elsaleh, P. Barnaghi, and S. Meissner, “An internet of things platform for real-world and digital objects,” Scalable Computing: Practice and Experience, vol. 13, no. 1, 2012.

[26] Linked Open Vocabularies, http://lov.okfn.org/dataset/lov/about

[27] Linked Open Vocabularies for Internet of Things, http://www.sensormeasurement.appspot.com/?p=ontologies