

An extensible and active semantic model of information organizing for the Internet of Things

Yunchuan Sun · Antonio J. Jara

Received: 28 February 2014 / Accepted: 29 April 2014 / Published online: 9 August 2014
© Springer-Verlag London 2014

Abstract Data management and information processing play the key roles in developing the Internet of Things (IoT). The requirements of a well-defined data model for IoT involve in six aspects: semantic supporting, active data extracting and explaining, flexibility and extensibility, enabling to manage massive and heterogeneous data, supporting formal organization, and solid mathematic-based theory. This paper aims to exploring an extensible and active semantic information organization model for IoT to meet the above requirements, and the primary idea is “Object-cored organizing data, event-based explaining data, and knowledge-based using data.” The proposed model involves two layers: the object layer and the event layer, and both of them are discussed in detail including conceptions, schema definitions, and the rule-based knowledge representation. Semantic reasoning can be supported by the knowledge base which involves in a set of reasoning rules on semantic relations among objects or among events correspondingly.

1 Introduction

In the long evolving history of human kind, creating and inventing new tools and things are the primary activities for the purpose of seeking more comfortable lives. Information exchanging and knowledge sharing play the key roles

during the process of tools innovation. The advent of the information technology, especially the Internet, leads to a new era of information exchanging and knowledge sharing. Data, information, and knowledge are the constant themes during the development of the information and communication technology (ICT). In recent years, an exciting and even puzzling paradise of data is emerging into our life with the development of the Internet of Things (IoT). People would lose their way to master the valuable information in massive and heterogeneous data captured by trillions of smart devices in IoT. Volumes of data are continually generated with the changes in the physical world from time to time. There is a list of primary challenges on data, information and knowledge issues in IoT: How to organize and manage the massive and heterogeneous data streams, how to extract meaningful and useful information from the raw data, how to validate the quality of data (QoD), how to acquire knowledge—in terms of models and behaviors insight—automatically from the massive information, how to annotate and describe semantically the data, and how to execute reasoning based on the existed information and knowledge automatically and intelligently. The most important in all these problems is to build a reasonable integrated model to organize and manage data, information, and knowledge in IoT.

In the traditional computing and networking systems, there have been a lot of successful efforts in the areas of managing and processing of data, information, and knowledge. The traditional relational data model, which organizes the data into a set of formal tables, has laid the foundation for the current information technologies featured by its widely applications in industries nowadays and has been regarded as the most distinguished one in all data management models [1, 2]. It is difficult to image that what the current cyber-physical world would

Y. Sun (✉)
Business School, Beijing Normal University, Beijing, China
e-mail: yunch@bnu.edu.cn

A. J. Jara
University of Applied Sciences Western Switzerland (HES-SO),
Delémont, Switzerland
e-mail: jara@ieee.org

be if the relational data model had not been there. Other traditional data management models, such as *hierarchical model* (organizing the data with a tree structure) [3, 4], *network model* (organizing data using records and sets with a network structure) [5, 6], and *Object-relational model* (organizing the data with a object-oriented way) [7], are also known data models and have been widely used in some specific temporary phases or in some specific fields. *Entity-relationship model* is an abstract conceptual data model to represent structured data [8], which has been widely used in database modeling. No-SQL models developed in recent years aim to dealing with the poor performance of the traditional relational data model on certain data-intensive applications such as social applications of Web 2.0, such as Facebook, twitter, and Google [9, 10], which supports weak guarantees of the data consistency. No-SQL databases are categorized according to the way they store the data and fall under categories such as Key-Value models [11], BigTable implementations [12, 13], document store databases [14], and graph databases [15]. Big Data is a new term occurred in recent years [16, 17] and is regarded as the potential driver of the future economics. Indeed, Big data in different applications is organized with different data models.

Actually, in most of the current Web applications, the semantic meaning behind detail raw data stored in various kinds of databases is often explained by independent application layers. Only the professionals, such as the designers and developers of the database and application systems, can grasp the ropes of the translating laws from data to information of semantic meanings, with the support of tedious developing documents.

The computers and smart systems cannot understand and explain the semantic of the data in an easy way. A series of technologies, such as XML (eXtensible Markup Language, www.w3.org/XML) and XML schema (www.w3.org/XML/Schema), RDF (Resource Description Framework, www.w3.org/TR/2004/REC-rdf-nt-20040210), and Semantic Web [18], have been proposed to deal with the semantic issues for machine understanding. Semantic network is a network model to represent knowledge by describing the semantic relations between conceptions [19, 20]. Semantic link network (SLN) is a more general model which manages semantic data with semantic nodes and semantic links and represents knowledge with reasoning rules [21–24]. The semantic nodes can be any resources, classes of resources, or even a semantic link network. The reasoning rules are for semantic reasoning. New semantic links may be derived out by rule reasoning. Semantic link network is a description of relations between objective existences rather than to represent fine human knowledge.

Though these traditional models have been widely used in industries, they cannot meet the requirements of the next

trend of the information technology. The requirements of a well-defined data model for the IoT involves in the following six aspects.

- *Semantic supporting* Fully intelligentization is the next step for the development of the IoT, also known as the *Semantic Web of Things* [25]. A semantic data model enables the understanding of the data captured by the sensing devices in the IoT. Semantic technologies will play the key role during the development of the IoT. Semantic interfaces are the most important technologies for realizing the interconnection and interactive communication among smart objects. Active reasoning with the support of enriched semantics is an essential module for the integration of knowledge and intelligence. A well-defined semantic data model is the primary foundation to support these semantic technologies.
- *Active data extracting and explaining* Based on enriched semantics, the semantic data model is required to have the ability to extract useful and refined information from the raw data in an active way, to understand the meaning of the data and to explain the meaning to applications or users automatically and actively.
- *Flexibility and extensibility* IoT will be the biggest interconnected system in the world which will be filled of varieties of things and will be changing and extending continuously. Trillions of smart things would be generating different types of data and transmit them to the Big Data platforms enabled by Cloud Computing infrastructure. A flexible and extensible data model is really important to manage and organize the various and growing data from the IoT and be able to exploit and extract all the data potential in order to enhance the existing daily processes in our works and personal lives.
- *Enabling to manage massive and heterogeneous data* The massive volume and heterogeneity are the primary features of the data in the IoT which is determined by the trillions and the varieties of things. It is a grand challenge to organize and manage these data in an efficient and effective way, especially to execute queries and operations in such a tanglesome context. A well-built data model is required to provide a quick and flexible solution to seek out the destinations relevant to the problems.
- *Supporting formal organization* A formal organization and format can assure an excellent way to data management. The characteristics of heterogeneity and complexity imply the challenge to organize and manage the data in a formal way.

- *Building on solid and mathematical foundation* A solid and mathematical foundation is necessary for a success data model. The strict forms and proved logic assure to avoid the possible faults and potential omissions of the data model. Relational algebra, formalized theory, and integrity theory are obvious examples to the relational database model. For example, emerging solutions are focused on Linked Data, in order to facilitate the mapping between different data models. Therefore, adequate complex networks and graph theory background is required to offer a suitable and integral solution.

An extensible and active data model is the key to develop the intelligent systems and to integrate different local applications in the IoT. The traditional data models such as relational data models, object-oriented models, and other models cannot meet these requirements of the IoT. Up to now, most of the existed local applications and demo systems of the IoT adapt traditional data models or simple plain texts. Even these applications are confined in a small range, however, there are some barriers in data managing and processing.

In details, the most extended for the IoT is the solutions based on the W3C Semantic Sensor Networks (SSN) incubator group [26]. The W3C SSN ontology comes with a wide range of relevant use-cases for the IoT, and this SSN ontology is one of the technologies that are much more extended in Europe for IoT, widely supported in projects and industrial solutions. Recently, it is being promoted for the interoperability among heterogeneous IoT resources in cloud-enabled solutions such as the presented in the OpenIoT EU platform (www.openiot.eu).¹

At the same time, several industrial-driven solutions are being also defined. The most extended is defined by the Open Geospatial Consortium (OGC). They have produced a wide range of standards for wireless location aware services, as well as sensor web services. OGC's standards cover from general and scalable models for describing sensors such as OGC-SensorML [27], as the standards for the exploitation through the integration with external systems through APIs and also for the query the capabilities of the sensors [28].

These approaches are mainly based on the resource description framework (RDF), which uses SensorML and/or SSN to define the classes which represent the concept of subjects, objects, and predicates. This means you can start making statements about classes of thing, and types of

relationship. RDF allows you to describe in human readable text the meaning of a relationship or a class.

RDF is exported in a number of file formats. The most common is RDF + XML and XSD. In addition to the relationship defined by RDF, OWL is used to add semantics to the schema. It allows you to specify far more about the properties and classes. It is also expressed in triples. For example, it can indicate that "If A isTogetherTo B" then this implies "B isTogetherTo A." Another useful thing OWL adds is the ability to say two things are the same one. This similarity feature is very helpful for joining up data expressed in different schemas. This last issue is highly important since, this bring the potential to build the relationship and joining up data from multiple sites (i.e., "Linked Data"). Thereby, facilitating the heterogeneous data streams integration.

Finally, even when RDF and OWL, using the mentioned SensorML and SSN ontologies bring a huge potential and capabilities; the industrial-driven solutions from oneM2M in collaboration with IPSO Alliance, OMA, 3GPP, ETSI M2M, ZigBee, among others are focused on OMA Web objects [25]. OMA Web Objects provides a simple binary representation of the resources in order to optimize communications performance.

Since the market disjunction and the needs for a holistic model that work into the existing and emerging semantic spaces.

This work provides an agnostic solution to satisfy the needs to develop an extensible and active semantic data model for the IoT, which supports massive and heterogeneous data managing, semantic reasoning automatically, information extracting actively, and extending flexibly. Such a data model can lay a theoretical foundation for researches and developments in the field of the IoT.

In this paper, we aim at developing an extensible and active two-layer semantic data model based on an extended model of semantic link network for the IoT.

2 Framework of the 2-layer model

Just like a people in the society, a smart thing in the IoT has its own brain (as a memory) to store data, information, and knowledge and (as a thinker) to make decisions according to the context information perceived with his sensory system for the next step. Of course, it is capable of communicating with other smart things or people to share data, information, and knowledge. Even more, it can deduce new knowledge from the volumes of experienced information. The centric server would communicate with individuals to pull information and knowledge from individuals, then integrate the sliced information and knowledge into the global ones, and finally distribute integrated information,

¹ OpenIoT: IoT Semantic Interoperability-Challenges, Best Practices, Solutions and Next Steps www.probe-it.eu/wp-content/uploads/2013/10/IERC-AC4-SemanticInteroperabilityManifesto-V1_130830-Final1.pdf.

knowledge, or commands to individuals. We should adapt a mechanism which supports both centric management and high degree of autonomy.

As discussed above, there will be large numbers of smart things in the IoT, even in the local applications of the IoT (named as Intranet of Things [29]), which link the physical space and the cyber space together. The states of the physical space are changing continually due to the continual events triggered by actions of the actuators or other outside condition changes. The smart devices capture these changes and generate detailed, complicated, and overloaded data of various types which should be submitted to the cyber space. Sensing devices, such as thermal sensors, optical sensors, humidity sensors, and other sensors, can feel the environment changes. RFID readers can locate the moving objects embedded with RFID chips. Big data is then generated by a large bulk of objects even in a small area of the IoT applications. It is impossible for users, unfortunately, to look into these detailed data manually for problem-solving. Generally, people try to find solutions based on not detailed data but “event” information intuitionally which is extracted from volumes of trivial data. Events extracted from the detailed and complicated data might be more meaningful and useful for users or decision systems to making decisions. Figure 1 shows an integrated cycle of data, information, and knowledge between physical world and cyber world. The gap between data and events should be bridged for the purpose of enabling people or decision systems to solve problems.

The primary missions for the semantic and active data model for the IoT include (1) organizing and managing the

data to describe the state of the objects and relations between them; (2) extracting and integrating event information actively from the detailed raw data; and (3) organizing and managing the events and relations between them.

The proposed model in this paper consists of two layers: the object layer and the event layer. The object layer would adopt an extended model of semantic link network to organize and manage the data with an object-oriented way where each object could be viewed as a node and semantic relations between objects could be viewed directed links with a semantic annotation in the network. The event layer also adopts the extended model of semantic link network to organize and to manage the event information where each event is a semantic node and semantic relations between events can be regarded as semantic links correspondingly. Each layer is with a knowledge base (a set of reasoning rules), respectively, for the purpose of finding implicit and potential and useful information by executing semantic reasoning. The active features lie in two aspects. Above all, the well-defined event schema in the event layer can be used to generate new events from the data in object layer actively. The reasoning rules stored in the knowledge bases can be used to seek out the useful and potential information such as implicit relations between objects or events automatically and actively. The schema for the proposed model involves in three modules: schema of object layer, schema of event layer, and the schema for knowledge bases as shown in the Fig. 2. The following sections discuss the modules of the architecture in detail.

Fig. 1 Cycle of data and information in the Internet of Things

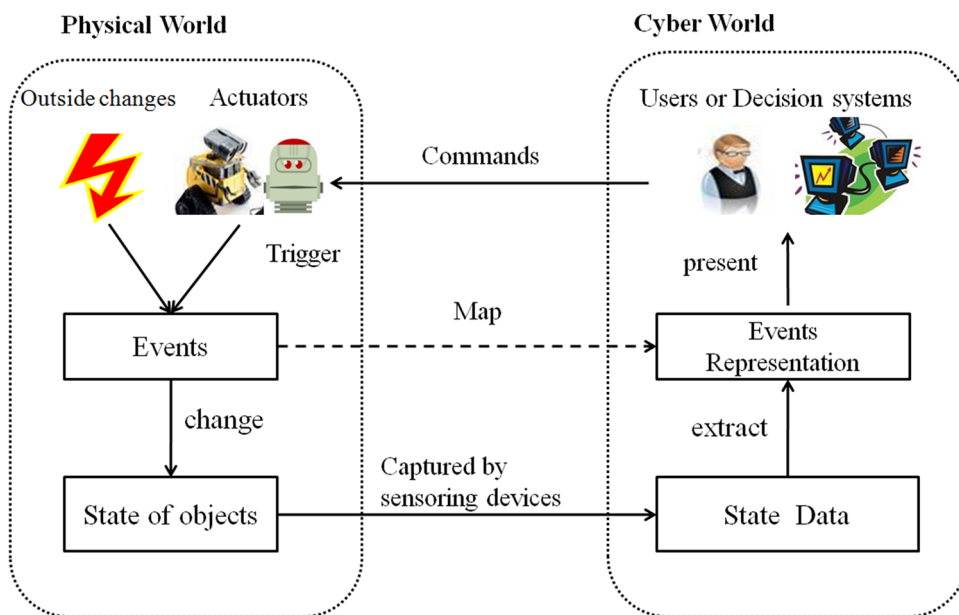
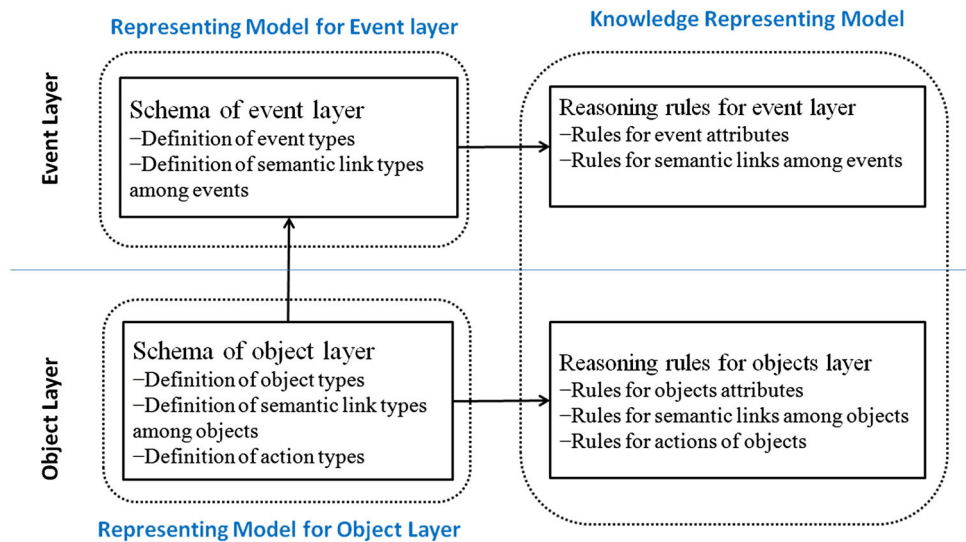


Fig. 2 The architecture of the proposed data model



3 Object layer

The object layer aims to represent the real world in an intuitionistic and exact way. The current state of the real world involves two types of contents. One is the state of physical objects, and the other is the semantic relations between different objects. The state of objects can be described with a list of attributes, such as temperature, location, size, or any other interesting features. These attributes would be selected according to domains and applications. Some of these attributes describe personal state of the objects, and others might describe the environment state which can be perceived by the objects. The semantic relations between objects are the most significant information of the state of the real world. The intelligent systems require the meaning of these relations to make decisions.

Therefore, the object layer should describe the above two types of state contents with a semantic, intuitive, and exact approach. This layer stores the raw data. With the continuing state changing of the physical world, the data stored in this layer will change continually. In the IoT, the sensing devices capture the state information in time and submit the information via the networks to the data center.

Moreover, smart objects would be endowed of some given abilities to make actions for the purpose of making response to the physical world by commands from information world which would be made according to patterns (knowledge) discovered from analyzing captured data or provided by domain experts.

3.1 Semantic link network

Semantic link network (SLN) is a semantic data model for managing resources and their semantic relations. It focuses on representation and managing the semantic relations

between objects and addresses on the reasoning among relations based on domain rules [2].

Knowledge plays a primary role in intelligent systems. Reasoning rules are generally used to represent the knowledge in the model of semantic link network. Reasoning rule is one of the most efficient ways to describe knowledge and to execute reasoning in intelligent systems. Due to different types of elements, the forms of the rules in two layers are different. There are four rule types in object layer and three types in event layer. In this section, we discuss these reasoning rules in both layers respectively. Firstly, we should introduce some basic conceptions about the semantic links before moving on.

- Equivalent link, denoted as e . Resources R and R' are called equivalent if they point to the identical resources or backups of the same resources. Obviously, a resource is equivalent to itself.
- Unknown link, denoted as $null$. If the semantic relation between two resources R and R' is unknown or implicit, we say there is a $null$ link between R and R' . The $null$ link does not mean there is no semantic relation and can be replaced with a new link type once the relation is determined.
- Empty link. If there is no any semantic relation between two resources R and R' , we say there is an empty link, denoted as $R \xrightarrow{\phi} R'$, and the semantic link type is ϕ , respectively.
- Inverse link. For two resources R and R' , if there is a semantic link from R and R' with type α , its inverse semantic link is defined as the semantic link from R' and R with type α^{-1} . That is, the inverse of the semantic link $R \xrightarrow{\alpha} R'$ is $R' \xrightarrow{\alpha^{-1}} R$.
- Complement link. If there is no semantic relation α between resources R and R' , we say that there is a

complementation of α between R and R' , denoted as $R \xrightarrow{\neg\alpha} R'$, and the semantic link type is $\neg\alpha$, respectively.

This links build a *Linked Data network* through a formal way. Thereby, this facilitates the reasoning and validation of the mappings.

3.2 Object semantic link model

We can extend the SLN model by attaching attributes on nodes to describe their states with a semantic way. The semantic link network is called Object SLN (OSLN) for its nodes are all objects in the IoT.

For each smart object in the Internet/Intranet of Things, its current state can be described as a list of attribute values with the form of (a_1, a_2, \dots, a_n) , where each a_i stores the data about one aspect of the object state. Different objects may have different attributes lists. The semantic meaning of each a_i can be declared by the standard schema definition in a specific domain.

For each pair of smart objects O_1 and O_2 in the IoT, their relations can be described by semantic links as $O_1 \xrightarrow{linkType} O_2$, where *linkType* is a possible semantic relations between O_1 and O_2 . Herein, the semantic meaning of *linkType* can be also declared and explained by the standard schema definition in the given domain. For example, we can use $mobile_1 \xrightarrow{sameOwner} computer_2$ to state that two different devices $mobile_1$ and $computer_2$ be owned by the same owner, where *sameOwner* is a predefined relation.

Thus, the state of Internet/Intranet of Things can be represented with a 2-tuple (*objectSet*, *linkSet*), where *objectSet* is a set of objects and each of which carried a list of attribute values, and *linkSet* is a set of semantic links to describe the semantic relations between objects in the context of IoT. Such a 2-tuple model can meet the requirements of the object layer in the following aspects.

- *Enriched semantics* Objects can be described by the given set of attributes, each of which represents an aspect. The semantic meaning of the attribute can be explained by the schema definition (see Sect. 4.2), and it can also be understand by the machines. The semantic relations between objects are explicitly represented and organized with semantic links in the model.
- *Intuitively* The model provides an intuitive way to describe the world filled with objects and their relations compared to the traditional data base models and the No-SQL models.
- *Easy to understand, easy to explain, and easy to use* Such an intuitive way is easy to understand by the non-technique and non-professional users. The known

social network LinkedIn (www.linkedin.com) adopts similar technology to describe the relations between persons.

3.3 The schema definition for object layer

There would be thousands or even trillions of things in the future Internet/Intranet of Things, even in local ones. These things can be classified into different types according to their functions, especially in the Intranet of Things. Actually, in most of cases, we build Intranets of Things for given domain applications first and further integrate them into the IoT. The objects in the context of Intranet of Things can be grouped into several types, and each type of objects might have the same list of attributes for their identical functions, i.e., there is a set of different objects types for an Intranet of Things. Meanwhile, the possible semantic relations between two objects can be determined by their object types. For example, the possible types of a semantic link between a mobile phone and a computer might be “have the same owner,” “have the same brand,” “connecting,” “have the same color,” “ neighbored to,” or any other possible relationships. We should define a set of the semantic link types between a pair of objects types for a given Intranet of Things for the convenience. Furthermore, in the context of Internet/Intranet of Things, some smart objects are capable of making actions according to some provided conditions or commands from the centric server. For example, smart air-conditioner can automatic switch or adjust itself to keep the temperature at expected according to the auto-detected identifiers of the people in the room whose required comfortable temperature have been discovered from history data. A kind of objects would have a set of similar functions.

While we decide to build an Intranet of Things for a certain application, the object types, link types, and action types should be defined firstly. Such a set of object types as well as the set of semantic link types and the set of action types form the schema of the OSLN for the Intranet of Things.

The schema of an OSLN can be denoted as 3-tuples (*OT*, *SLT*, *AT*), where *OT* is the set of object types, *SLT* is a set of semantic link types and *AT* a set of action types in view of the object types in *OT*. Each object type is defined by a type name and a list of attributes; also, each attribute includes a name, range domain, semantic label, and the instruction. The range domain defines a range for the attribute value; the semantic label is used to indicate the semantic explanation of the attribute and to support information exchange with a standard and semantic way, and the instruction provides the detail information of the attribute in the context.

The object types schema for the object layer can be denoted as $OT = \{ot_1, ot_2, \dots, ot_m\}$, where each ot_i is an object type and each ot_i can be defined with a lists of attributes $\{Oid, a_1, a_2, \dots, a_k\}$. Each attribute is used to indicate an aspect of the object by a definition with four components:

- Attribute name.
- Domain, to indicate the domain of the attribute.
- Semantic label, to describe the semantic meaning of the attribute.
- Instructions, to describe the requirements or other instructions of the attribute.

For example, the definition of the attribute *Oid* is (**Attribute name:** *Oid*, **Domain:** *ID domain*, **Semantic label:** *Identifier of object*, **Instructions:** *Oid is a unique and required identifier for each object*).

The second component of the schema is a set of semantic link types grouped by the pairs of object types, which defines all possible semantic link types in the context. For each pair of object types (ot_i, ot_j) , the set of all possible semantic link types is denoted as $[ot_i, ot_j]$. Thus, the set of all semantic link types in the Intranet of Things, denoted as SLT , can be represented as $SLT = \cup_{i,j}[ot_i, ot_j]$ for all i, j .

Each semantic link type, $lt \in [ot_i, ot_j]$ aims to describe a type of the semantic relations from the start object type ot_i and the end object type ot_j . For example, between mobile and computer, $\{sameOwner, sameBrand, sameColor, connectingTo, neighborTo\}$ might be the possible set of semantic link types. Of course, we can trade off the set according to the context. For each semantic link type, we should define its name, the start object type, the end object type, semantic label. If necessary, semantic link can also be allowed us to have some attributes for detail description, e.g., *neighborTo* might have an attribute *distance* to declare the distance between two objects. Thus, the definition of a semantic link type consists of five components:

- Name.
- Object pair, to confirm the start and the end object type for the semantic link type.
- Semantic label, to indicate the meaning of the semantic link type.
- Attribute, to describe some certain aspects of the semantic link in detail.
- Instruction.

The third component of the schema is a set of action types grouped by the object types. For each type of objects, ot_i , which need abilities to make actions, the set of action types can be denoted as $[ot_i]_A$. So the set of all action types

in the Intranet of Things can be represented as $AT = \cup_i [ot_i]_A$ for all i .

Each action type $at \in [ot_i]_A$ aims to describe how the action will change the state. Of course, there might have some kinds of smart objects which have no function of action because they are only sensing devices. For these types of objects, the set of action types is ϕ .

In all, the schema for object layer can be represented as (OT, SLT, AT) , where each object type has a definition of name, domain, semantic label, and instruction, and each semantic link type has a definition of name, start object type, end object type, and semantic label.

We can scheme out a schema for the Intranet of Things according to the given applications in line with the above schema model firstly. Then, the data about the objects and their relations in the context can be inserted into the instance systems. Different Intranets of Things for the same or similar kind of applications can share one schema or a revised one. It is the most convenient for information exchanging and knowledge sharing with a semantic way among intranets supported by a standard schema. Different schemas defined for Intranet of Things of different enterprises in a business stream might share a schema partly, for example, object types and semantic link types. With the help of the semantic labels described with a standard form, it can also facilitate the information exchanging and knowledge sharing to a great extent.

The schema for an Intranet of Things is open and extendible. Users can easily add a new object type by inserting the corresponding information, while some new device joins the intranet, or updating the object type while some kind of devices changes functions. Similarly, the semantic link types and action types can also be revised according to the changing situation. There are two different means to extend the schema. We can update the schema with a well-designed way before installing or removing a bulk of devices of specific types to the Internet/Intranet of Things. On the other hand, in some occasional cases, the application systems can update the schema with an unplanned way by adding new types of objects, semantic links, or actions after some new devices had been installed to the Internet/Intranet of Things.

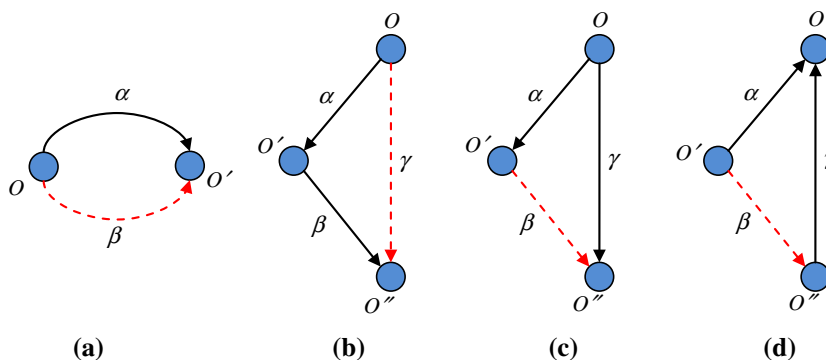
3.4 Reasoning rules for object layer

In the object layer, semantic reasoning involves in semantic links between objects, attributes of objects, and actions of objects. We can get four groups of the reasoning rules according to the involved elements.

Reasoning rules among semantic links.

Reasoning rules between two objects.

Fig. 3 Basic forms of the reasoning rules among semantic link types (Solidline arrow: existed links, dashed arrow: deduced links)



Rule Form 1 For two semantic link types α and β , $\alpha \Rightarrow \beta$. If there is a semantic link $O \xrightarrow{\alpha} O'$, then there is a certain semantic link $O \xrightarrow{\beta} O'$, as shown in Fig. 3a.

A reasoning rule with the form $\alpha \Rightarrow \beta$ means that α implies β , or α is stronger than β , also denoted as $\beta < \alpha$. We are inclined to replace a weak semantic link with a stronger one if possible.

Characteristic 1 The intense relation between semantic link types is symmetrical, non-reflexive, and transitive, and is a relation of partial order.

Characteristic 2 For any semantic link type α , we have $null < \alpha$. That means, $null$ is the bottom in any partial order relations.

Rule Form 2 For three semantic link types α , β , and γ , $\alpha \cdot \beta \Rightarrow \gamma$. If there are two semantic links $O \xrightarrow{\alpha} O'$ and $O' \xrightarrow{\beta} O''$, we can get $O \xrightarrow{\gamma} O''$ from the rule, as shown in Fig. 3b.

Rule Form 3 For three semantic link types α , β , and γ , $\gamma^{-1} \cdot \alpha \Rightarrow \beta$. If there are two semantic links $O \xrightarrow{\alpha} O'$ and $O \xrightarrow{\gamma} O''$, we can get $O' \xrightarrow{\beta} O''$ from the rule, as shown in Fig. 3c.

Rule Form 4 For three semantic link types α , β , and γ , $\alpha \cdot \gamma^{-1} \Rightarrow \beta$. If there are two semantic links $O' \xrightarrow{\alpha} O$ and $O'' \xrightarrow{\gamma} O$, we can get $O' \xrightarrow{\beta} O''$ from the rule, as shown in Fig. 3d.

Characteristic 3 There are some general reasoning rules listed as follows:

1. $\alpha \cdot null \Rightarrow null, null \cdot \beta \Rightarrow null$;
2. $\alpha \cdot \phi \Rightarrow null, \phi \cdot \beta \Rightarrow null$;
3. $\alpha \cdot e \Rightarrow \alpha, e \cdot \beta \Rightarrow \beta$.

Lemma 1 For three semantic link types α , β , and γ , we have

1. $\alpha \cdot \beta \Rightarrow \gamma$ is equivalent to $\beta^{-1} \cdot \alpha^{-1} \Rightarrow \gamma^{-1}$.
2. $\alpha^{-1} \cdot \beta \Rightarrow \gamma$ is equivalent to $\beta^{-1} \cdot \alpha \Rightarrow \gamma^{-1}$.
3. If $\alpha \Rightarrow \alpha'$ and $\alpha' \cdot \beta \Rightarrow \gamma$, then $\alpha \cdot \beta \Rightarrow \gamma$.

Attributes are used to describe the semantic content of the objects. From the interrelations between attributes of different objects, potential and useful information between objects can be deduced. Several forms of reasoning rules can be used to represent these interlinks among attributes and enable the intelligent systems to execute the reasoning automatically.

Rule Form 5 $O \cdot (a_1 = A) \Rightarrow O \cdot (a_2 = f(A))$, where A_1 and A_2 are two attributes of the object O , A is the value assigned to a_1 and f is a function to reflect the interrelation of a_1 and a_2 .

Rule Form 6 $O \cdot (a = A) \wedge O' \cdot (a' = A') \wedge \theta(A, A') \Rightarrow O \xrightarrow{\alpha} O'$, where a, a' is O, O' , respectively, A, A' is the value assigned to a, a' , respectively, θ is a binary relation and α is a semantic relation between objects O and O' .

Rule Form 7 $O \cdot (a = A) \wedge O \xrightarrow{\alpha} O' \Rightarrow O' \cdot (a' = A')$, where a, a' is attribute of the object O, O' , respectively, A, A' is the value assigned to a, a' , respectively, and α is a semantic relation between objects O and O' .

Rule Form 8 $O \xrightarrow{\alpha} O' \Rightarrow O \cdot (a = A)$, where a is an attribute of the object O , α is a semantic relation between objects O and O' and A is the value assigned to a .

Rule Form 9 $O \cdot (a = A) \Rightarrow O \xrightarrow{\alpha} O'$, where a is an attribute of the object O , α is a semantic relation between objects O and O' and A is the value assigned to a .

4 Event layer

The state data layer stores the data about the current state of the Internet (or Intranet) of Things. For the purpose of keeping consistent with the continuing changing physical

world, the semantic link networks in the data layer should be modified in time. The evolving progress might be the most significant for the business of the enterprises to discover potential useful patterns and new knowledge which might be beneficial to new business models and possible opportunities. Unfortunately, similar to traditional data models, the data layer cannot record the constantly changing process of the state while it addresses only on recording of the current state. A two-layer semantic model is proposed to store and process the evolving process with semantic link network models in [3]. With the support of the conception of the event semantic link network, we can develop a model to record the evolving history of the Internet/Intranet of Things using the idea of event semantic link network.

In this section, we will firstly propose some conceptions on events and then the primary idea of semantic link network of events. Finally, the uniform of the schema definition for event information layer is developed.

4.1 Basics of event

In the physical world, the state will change only when some events occur. The event might be caused by outside conditions such as changes of the environment or by actuators controlled by human beings or automatic agents. Correspondingly, the semantic link network should be modified in the virtual space by updating the objects or their relations for the purpose of seamlessly integrating the two worlds. Namely, an event means a set of objects or a set of semantic relations should be updated in the cyber space concurrently. In this light, an event can be regarded as a set of operations on the objects or semantic relations between objects in the semantic link network of state data layer.

Definition 1 An event on an SLN S , denoted as $E = \{p_1, p_2, \dots, p_n\}$, is a set of operations of the following primary types which are interlinked on the same topic.

1. Appending an object O with the list of attribute values, denoted as $A(O)$;
2. Deleting an object O , denoted as $D(O)$;
3. Updating an object by modifying the attribute values, denoted as $U(O(a_1, a_2, \dots, a_k))$;
4. Adding a new semantic link l , denoted as $A(l : O \xrightarrow{\alpha} O')$;
5. Deleting a semantic link l , denoted as $D(l : O \xrightarrow{\alpha} O')$; and
6. Updating a semantic link $l : O \xrightarrow{\alpha} O'$ by replacing its semantic link type α with β , denoted as $U(l : O \xrightarrow{\beta} O')$.

Theoretically, any set of some operations can be viewed as an event. However, not all of these theoretical events are

useful or interesting in practical applications for the reason that some operations in the same set may be irrelevant or even incompatible. How to organize the diversity of operations to form events is the key issue. The followings are some criterions for an event.

1. *Temporal* All operations involved in an event should occur in the same time sect.
2. *Topic oriented* All operations involved in an event are related to the same topic.
3. *Consistency* Two conflicted operations cannot be included in the same event.
4. *Integrity* An event should not include the following types of operations.
 - a. Impossible operations, such as update or delete a non-existed resource or a semantic link, append a semantic link which start resource or end resource is not in the SLN.
 - b. Redundant operations, such as two repeat operations, or two contrary operations, for example, $A(R_0)$ and $D(R_0)$, or $D(l_0)$ and $A(l_0)$ in the same event are redundant operations.

Of course, these criterions are not indispensable to forming an event. We can organize freely any set of operations into an event according to the actual demand.

An event is called atomic if all its operations either all occur, or nothing occurs. In other words, an atomic event means indivisibility and irreducibility. An event is called composable if its operations can be divided into several subgroups each forms an event. An event is called loosely composable if its operations can be freely shrinkable or extendible. For example, a meeting event can be composed by a set of registers which number is not fixed. We have the following characteristic about events. For the convenience, we use ϕ to denote the empty event which does not involve in any operations.

1. An event involved in only one primary operation is of course atomic.
2. All operations in an atomic event are concurrent.
3. The intersection of any two atomic events is empty.
4. An event might consist of one or more atomic events.
5. Different operations in one event might occur in different areas or over a long time span.
6. Some implicit events might be caused in the dark by other events.

Event types should be predefined by domain experts or abstracted from volumes of frequent occurred patterns automatically. Indeed, Events can be extracted from detailed and large scale data based on the event schema which consists of a list of predefined event types. An event type can be described as a 2-tuple (*Precondition*, *EventDescription*),

where *Precondition* is a logic expression to determine the event and *EventDescription* is a description of the event including event name and other information. Such a mechanism can efficiently grasp the critical data for the useful events by filtering the detailed data with the well-defined event types in the schema. Generally, the event schema can be worked out by the domain experts or can be discovered from existed massive event warehouse with automatic and intelligent mining tools.

4.2 Event semantic link network

In the evolving process of the semantic link network, there might be numerous of events. Among these events, there are kinds of inherent semantic relations such as sequence, cause-effect, inclusion, concurrency, and any other possible relations between events. Each event can be viewed as a node. Events on a semantic link network and their relations would form an event semantic link network (*ESLN*) which records the evolving of the original object semantic link network. According to the original state of a semantic link network and its *ESLN*, we can reconstruct the evolving process of the semantic link network.

For each event, a list of attribute values (a_1, a_2, \dots, a_n) can be used to describe its information, where each a_i reflects one aspect of the event information. Different types of events may have different attributes lists. The semantic meaning of each a_i can be declared in the standard schema definition for a given application.

For each pair of events E_1 and E_2 in the IoT, their relations can be described by semantic links like $E_1 \xrightarrow{elinkType} E_2$, where *elinkType* is a possible semantic relations between E_1 and E_2 . Herein, the semantic meaning of *elinkType* can be also declared and explained in the schema definition in the given domain.

Thus, an event semantic link network *ESLN* can be represented as a 2-tuple (*eventSet*, *eLinkSet*), where *eventSet* is a set of events and *eLinkSet* is a set of directed links to indicate the semantic relations between events.

Features of the event semantic link network are listed as follows.

- *Enriched semantics* The semantics can be assured by two aspects: (1) event semantics are described by the attributes list and semantic annotations, and (2) relation semantics are represented by the semantic links between events.
- *Unidirected expanding* Appending is the primary operation in the event semantic link networks. It would be rare to delete or to update an event in the context of event SLN. From this view, the event SLN will be growing continually.
- *Evolution recording* Event SLN is enabled to reflect the evolving process of the state of the physical world by

organizing the changing data with an event-oriented way.

- *Events stream in time* Each event has a timestamp as a default attribute to record its coming up. An events stream in time sequence can be formed by all events. Potential and useful patterns could be mined from the events stream.

4.3 Schema definition of event layer

An event is a set of data resulted from a list of correlated changes of the state in the physical space. Different event types describe different types of state changes. An event type involves in some object types, semantic link types, and detailed information of state changes.

The schema of the event semantic link network can be viewed as a 2-tuple $ES(ET, ELT)$, where *ET* is a set of event types, and *ELT* is a set of event semantic link types. Events can be classified according to the actual business or applications. An event type defines a set of operations and the type of each operation for event instances. The event types schema for the event information layer can be denoted as $ET = \{et_1, et_2, \dots, et_m\}$, where each et_i is a name of an event type and each et_i consists of a sequence of the primary operations in definition 1 and can be denoted as $et_i = \{p_1, p_2, \dots, p_k\}$, where each operation p_i may have some parameters of typed objects or typed semantic links. Meanwhile, et_i also defines the order of the operations. For each operation p , it takes one of the following forms:

1. $A(O)$, where O is of a given type ot ;
2. $D(O)$, where O is of a given type ot ;
3. $U(O(a_1, a_2, \dots, a_k))$, where O is of a given type ot ;
4. $A(l : O \xrightarrow{\alpha} O')$, where O, O' are of given types ot, ot' resp., α is a semantic link type;
5. $D(l : O \xrightarrow{\alpha} O')$, where O, O' are of given types ot, ot' resp., α is a semantic link type;
6. $U(l : O \xrightarrow{\alpha} O')$, where O, O' are of given types ot, ot' resp., α is a semantic link type.

Thus, an event type et can be defined as the following form.

$$Event\ Type: = \{$$

$$[A(O:ot); [A(O:ot); [...[A(O:ot)]]]];$$

$$[D(O); [D(O); [...[D(O)]]]];$$

$$[A(l:lt); [A(l:lt); [...[A(l:lt)]]]];$$

$$[D(l); [D(l); [...[D(l)]]]];$$

$$[U(O:ot); [U(O:ot); [...[U(O:ot)]]]];$$

$$[[U(l:lt); [U(l:lt); [...[U(l:lt)]]]]];$$

$$[a\ list\ of\ constraints\ on\ operations].$$

$$\}$$

where the explanation for each component in the format are listed in the Table 1.

There might be kinds of semantic relations between two events and the types of these semantic relations depend on the type of the start event and the end event. For an Internet (Intranet) of Things, it is necessary to define potential semantic link types between event types. These definitions of the semantic link types form the schema of event link types as the second component ELT in $E\hat{S}$. The semantic link types can be divided into two groups: the general (domain independent) link types, denoted as G , and the domain/application-oriented link types, denoted as D . In general, the following primary types of semantic links between events can be applied in any domain or any application.

1. *Equivalence* Two equivalent events E and E' , denoted as $E \xrightarrow{e} E'$, where E and E' are two events of the same type.
2. *Inclusion* An event E includes another E' , i.e., $E \supseteq E'$, denoted as $E \xrightarrow{include} E'$. Clearly, equivalence relation is a special case of inclusion; empty event (ϕ) is included in any event.
3. *Cause-effect* The occurrence of an event E' results in another E , denoted as $E \xrightarrow{ce} E'$.
4. *Sequence* An event E occurred before another E' , denoted as $E \xrightarrow{seqTo} E'$. Obviously, $E \xrightarrow{ce} E'$ implies $E' \xrightarrow{seqTo} E$.
- 5) *Concurrency* Two events E and E' occur on S concurrently, denoted as $E \xrightarrow{concurTo} E'$
- 6) *Connection* Two events E and E' are called connected if they share at least one resource or one semantic link, denoted as $E \xrightarrow{connectTo} E'$.

Table 1 Explanation for operation types of event types

Component	Explanation
$A(O: ot)$	Append an object O of type ot
$D(O)$	Delete an object O
$A(l: O \xrightarrow{\alpha} O')$	Append a link l of type α from O (of et type) to O' (of et' type)
$D(l: O \xrightarrow{\alpha} O')$	Delete a link l of type α
$U(O:ot)$	Update a resource O of type ot
$U(l: O \xrightarrow{\alpha} O')$	Update a link l of type α from O (of et type) to O' (of et' type)
Constraints on operations	Explanation for each constraint on operations, for example, the appended link in $A(l:l)$ should be related to some new resource
[...]	Option

Thus, we get that $G = \{e, ce, include, seqTo, connectTo, concurTo\} \subseteq ELT$.

Furthermore, we can define the domain/application-oriented *event semantic link types* according to the event types for the applications in the domain. An event semantic link type depends on the types of its start event and end event. The set of semantic link types between a pair of event types et and et' can be denoted as $[et, et']$. The set of event semantic link types for an Internet (Intranet) of Things is denoted as $D = \cup[et, et']$, for all pairs of event types. Of course, $D \subseteq ELT$, and $ELT=G \cup D$.

4.4 Reasoning rules for event layer

As discussed in Sect. 4.3, there is a set of general semantic link types which can be applied in any domain or any applications, denoted as $G = \{e, ce, include, seqTo, connectTo, concurTo\} \subseteq ELT$. According to the definitions and implications of these semantic link types, we can easily derive a set of reasoning rules related to these semantic link types shown in the Table 2.

Furthermore, we can define the domain-oriented *event link types* according to the event types for the applications in the domain. Similarly, we can work out the additional reasoning rules for the set of all event semantic link types including the domain-oriented and the general. Additionally, the reasoning mechanism relies not only on the reasoning rules between semantic link types, but also on the operational objects of the event instances. In light of this view, we should build a new reasoning mechanism according to the actual applications.

5 Discussion, conclusion, and future works

5.1 Discussion and conclusion

In our previous works, event-linked network (ELN) model is proposed in [30, 31] to regulate events and their internal semantic relations. A framework on constructing event-linked network from raw detail data is presented in [30]. We introduce how to organize sensing data with ELN model and illustrate how to execute efficient queries on ELN in [31]. Based on these works, this paper proposes an integrated model for data management, information extracting, and knowledge management in the context of the IoT. The object layer can represent the objects and their relations with the model of semantic link network and a reasoning rule set is attached in this layer to support the semantic reasoning among objects. The event layer can extract the event information from the raw detail data generated from the object layer and regulate the events and their relations using the event semantic link network

Table 2 Reasoning rules for the event semantic link network

No.	Rules	Explanation
1.	$eX \rightarrow X, X \cdot e \Rightarrow X (X \in ELT)$	Equivalent events share the same relations with other events
2.	$ce \cdot ce \Rightarrow ce$	If an event E causes another event E' and E' causes E'' , then E is also the cause of E''
3.	$seqTo \cdot seqTo \Rightarrow seqTo$	If an event E is the sequence of another event E' and E' is the sequence of E'' , then E is also the sequence of E''
4.	$ce \Rightarrow seqTo$	An event E is the cause of another E' , and then it occurs before E' .
5.	$include \cdot include \Rightarrow include$	If an event E includes another event E' and E' includes E'' , then E includes E''
6.	$e \Rightarrow concur$	Equivalent events occur concurrently
7.	$ce \cdot seqTo \Rightarrow seqTo$	If an event E causes another event E' , and E' is sequent to E'' , then E is also the sequence of E''
8.	$seqTo \cdot ce \Rightarrow seqTo$	If an event E' is the sequence of E , and E' causes another event E'' , then E is the sequence of E''
9.	$include \cdot ce \Rightarrow ce$	If an event E includes E' , and E' causes another event E'' , then E is a cause of E''
10.	$ce \cdot include \Rightarrow ce$	If an event E causes E' , and E' includes another event E'' , then E is a cause of E''
11.	$include \cdot seqTo \Rightarrow seqTo$	If an event E includes E' , and E'' is a sequence of E' , then E'' is a sequence of E
12.	$seqTo \cdot include \Rightarrow seqTo$	If an event E' is a sequence of E and E' includes E'' , then E'' is a sequence of E
13.	$include \cdot connectTo \Rightarrow connectTo$	If an event E includes E' and E' connects to E'' , then E connects to E''
14.	$concur \cdot concur \Rightarrow concur$	If an event E and E' occur simultaneously, and E' and E'' occur simultaneously, then E and E'' occur simultaneously
15.	$seqTo \cdot concur \Rightarrow seqTo$	If an event E' is a sequence of E and E' and E'' occur simultaneously, then E'' is a sequence of E
16.	$concur \cdot seqTo \Rightarrow seqTo$	If events E and E' occur simultaneously, and E'' is a sequence of E' , then E'' is a sequence of E

model. A reasoning rule set among events and their relations is also necessary for the purpose of intelligent reasoning. There is a list of advantages for such an integrated architecture.

At first, it is a perfect architecture which ingrates data, information with knowledge. The data layer can store not only the raw object-oriented data captured from the physical world through sensing devices or readers, but also the semantic relations between objects. The event layer manages the event information extracted from the data layer. The reasoning rules in the two layers can be used to find the new semantic links among objects and among events. Secondly, it records the evolving process of the physical world with the help of the event network. The semantic relations between events would be useful to discover the evolving patterns of the physical world and to predict the future trend. Thirdly, the proposed framework supports the information exchanging and knowledge sharing among Intranets of Things built for interconnected businesses. Additionally, it is easy to exchange the data between traditional relational databases with the data layer of the proposed model. The representations of the objects in the physical world and kinds of relations between these objects as well as the events in the information level and the internal relations between events are all semantic based. The enriched semantics with two layers can meet the requirements of the applications in the IoT. Furthermore, the model with enriched semantics could meet the requirements of various kinds of applications. Indeed, with

such a two level model, it is easy to quickly locate all objects which involved in a given event and to query all events which include a given objects vice versa. The mapping mechanism between the two levels provides a convenient and powerful platform to support kinds of applications.

Solid mathematic basis can assure the correctness and efficiency of a model. The schema theory proposed in [24] is a solid theoretical basis for Semantic Link Network to regulate its nodes and links in a normal form and provides efficient reasoning algorithms to deduce implied semantic relations between nodes. It also provides a powerful tool to build and manage an instance of semantic links. The proposed model in this paper can be also supported by of the schema theory in both levels of objects and events. Besides, a mapping mechanism theory from the algebra view would be built to support the mechanism between the mentioned levels.

Meanwhile, there are challenges for such an integrated model of data, event information and reasoning rules. First of all is the uncertainty of the event schema. Indeed it is impossible to predefine all possible types of events and some of them could only be defined once an event with such a type could have been occurred. So the schema can only deal with the existing event types. Also, it is difficult to make clear all internal complex relationships between events. Some events might be components of other big events. The nested and inclusive relations between events would be difficult to manage and process.

5.2 Future works

In the coming future, we would like to pay more attention on the following aspects. Firstly, we will try our best to apply the proposed extensible and active model in some industries of the IoT. Even more, we will develop a completed management system of the proposed model for the purpose of making it easier for future potential applications. Secondly, the internal relationships between different events will be further studied. On this basis, a theory of event algebra, including the mathematical definitions of the events and operations of events such as union, discourse, difference, components and etc., could be developed. Such an event algebra theory would be the foundation of the further schema theory and the normalized theory of the event layer and the algorithms of querying and reasoning. Finally, in the applications of large scales, the query and reasoning, especially the mapping between objects level and events level, would be one of the biggest challenges. For a given issue, the projection in the objects level of the involving events could narrow down the scale of the problem and then make the query and reasoning more efficiently. Vice versa, the projection in the event layer of a set of objects can be used to find the relative events, to trace back to the evolving of the objects and to mine the interesting evolving patterns. In a word, these incoming theories would provide useful ideas and efficient solutions for how to manage and how to use heterogeneous and mass data in the IoT.

References

- Codd EF (1970) A relational model of data for large shared data banks. *Commun ACM* 13(6):377–387
- Astrahan MM, Blasgen MW et al (1976) System R: relational approach to database management. *ACM Trans Database Syst (TODS)* 1(2):97–137
- Tsichritzis DC, Lochovsky FH (1976) Hierarchical data-base management: a survey. *ACM Comput Surv (CSUR)* 8(1):105–123
- Bjorner D, Lovengreen H (1982) Formalization of database systems: and a formal definition of IMS (Invited Paper). In: *Proceedings of the 8th international conference on very large data bases*, pp 334–347
- Bachman CW (1969) Data structure diagrams. *ACM SIGMIS Database* 1(2):4–10
- Bachman CW (1973) The programmer as navigator. *Commun ACM* 16(11):653–658
- Zhao L, Roberts SA (1991) Object-oriented data models: a perspective and comparative review. *J Inf Sci* 17(3):145–160
- Chen PP (1976) The entity-relationship model: toward a unified view of data. *ACM Trans Database Syst (TODS)* 1(1):9–36
- Peng D, Dabek F (2010) Large-scale incremental processing using distributed transactions and notifications. In: *The 9th USENIX symposium on operating systems design and implementation (OSDI 2010)*, Vancouver, BC, Canada
- Zhang C (2010) Supporting multi-row distributed transactions with global snapshot isolation using bare-bones HBase. In: *The 11th ACM/IEEE international conference on grid computing (Grid 2010)*, Brussels, Belgium
- Seeger M, Ultra-Large-Sites S (2009) Key-value stores: a practical overview. *Computer Science and Media*, Stuttgart
- Chang F, Dean J, Ghemawat S et al (2008) Bigtable: a distributed storage system for structured data. *ACM Trans Comput Syst* 26(4):1–4
- Wei W, Yu T, Xue R (2013) iBigTable: practical data integrity for BigTable in PUBLIC cloud. In: *Proceedings of the 3rd ACM conference on data and application security and privacy, CODASPY'13*, New York, NY, USA, pp 341–352
- <http://www.mongodb.com/document-databases>
- Angles R, Gutierrez C (2008) Survey of graph database models. *ACM Comput Surv (CSUR)* 40(1):1–39
- White T (2012) Hadoop: the definitive guide. O'Reilly Media. 3
- Snijders C, Matzat U, Reips UD (2012) 'Big Data': big gaps of knowledge in the field of Internet science. *Int J Internet Sci* 7:1–5
- Berners-Lee T, Hendler J, Lassila O (2001) The semantic web. *Sci Am* 284(5):34–43
- John F (1992) Sowa; semantic networks, encyclopedia of artificial intelligence. <http://www.jfsowa.com/pubs/semnet.htm>
- Quillian MR (1968) Semantic memory. In: Minsky M (ed) *Semantic information processing*. MIT Press, Cambridge, MA
- Zhuge H (2009) Communities and emerging semantics in semantic link network: discovery and learning. *IEEE Trans Knowl Data Eng* 21(6):785–799
- Zhuge H, Sun Y et al (2005) Algebra model and experiment for semantic link network. *Int J High Perform Comput Netw* 3(4):227–238
- Zhuge H (2007) Autonomous semantic link networking model for the knowledge grid. *Concurr Comput Pract Exp* 7(19):1065–1085
- Zhuge H, Sun Y (2010) The schema theory for semantic link network. *Futur Gener Comput Syst* 26(3):408–420
- Jara A, Olivieri A, Bocchi Y, Jung M, Kastner W, Skarmeta A (2014) Semantic web of things: an analysis of the application semantics for the IoT moving towards the IoT convergence. *Int J Web Grid Serv* 10(2/3):244–272
- Compton M, Barnaghi P, Bermudez L, García-Castro R, Corcho O, Cox S, Taylor K (2012) The SSN ontology of the W3C semantic sensor network incubator group. *Web Semant Sci Serv Agents World Wide Web* 17:25–32
- Botts M (2007) OGC implementation specification 07-000: OpenGIS sensor model language (SensorML)-open geospatial consortium. Tech. Rep
- Simonis I, Dibner PC (eds) (2007) OpenGIS sensor planning service implementation specification. *Implementation Specification OGC* pp 1–21
- Zorzi M, Gluhak A, Lange S, Bassi A (2010) From today's intranet of things to a future internet of things: a wireless-and mobility-related view. *Wirel Commun IEEE* 17(6):44–51
- Sun Y, Yan H et al (2014) Constructing the web of events from raw data in the web of things. *Mobile Inf Syst* 10(1):105–125
- Sun Y, Yan H et al (2014) Organizing and querying the big sensing data with event-linked network in the internet of things. *Int J Distrib Sens Netw* 1–11. Article ID 218521