

Research Article

Ontologies for the Transactions on IoT

Chao Qu,^{1,2} Fagui Liu,¹ and Ming Tao²

¹ School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China

² School of Computer, Dongguan University of Technology, Dongguan 523808, China

Correspondence should be addressed to Fagui Liu; fgliu@scut.edu.cn

Received 24 April 2014; Revised 4 September 2014; Accepted 7 September 2014

Academic Editor: Zhangbing Zhou

Copyright © 2015 Chao Qu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To facilitate the efficient use of information generated from the Internet of Things (IoT) by sensors and automatically drive entities to meet the needs of human beings, this paper presents an information processing framework using semantic web technologies. The framework represents the entities as semantic web services and automatically creates service sequences as transactions by analyzing the information need in specific queries. Moreover, the framework ensures the effective management and control of information collected by mobile sensors during the transaction. The foundation of this framework lies in a set of ontologies we proposed: goal ontology, role ontology, constraint ontology, message ontology, status ontology, space-time ontology, and activity ontology. In addition to their respective definitions, we also present a detailed exploration of the relationships between them. Finally, we validate the effectiveness of the framework and the ontologies with a case study of the logistics system. The evaluation shows that the ontologies support the framework well and our approach increases the logistics system efficiency with lower operating costs.

1. Introduction

The Internet of Things (IoT) vision [1] promotes the ubiquitous computing paradigm on a global scale. In ubiquitous and pervasive contexts, intelligence is embedded into objects and physical locations by means of a relatively large number of heterogeneous microdevices; for example, sensors are used in applications ranging from meteorology to medical care, environmental monitoring and security, and surveillance. This increase is accompanied by an increasing volume of data, as well as increasing heterogeneity of devices, data formats, and measurement procedures. It would be ideal if this information can be automatically extracted and processed in order to better support the current activity of users and satisfy their needs [2].

Semantic technologies can assist in managing, querying, and combining entities and observation data. Thus, it allows users to operate at abstraction levels above the technical details of format and integration instead, work with domain concepts and restrictions on quality. With the development of the semantic web [3], the use of ontologies is becoming common as a formalism to describe knowledge and information in a way that can be shared on the web is becoming

common. Ontologies currently have been widely used in knowledge engineering (KE) and artificial intelligence (AI) to structure the concepts of a domain. Ontology is represented as a knotted tree of linked concepts or classes. A class can have subclasses which represent concepts more specific than the super class. The properties of each concept describe the characteristics, the attributes of the concept, and the attributes restrictions. Ontology and the individual instances of the classes constitute a knowledge base.

The purpose of this paper is to present an information autoprocessing framework for IoT based on the semantic web technologies and describe the design of the top ontologies for the framework. The main contributions can be summarized as follows.

- (1) An effective framework is designed for automatically constructing and executing the transactions on IoT.
- (2) The designed ontologies are reasonable and meet the need of the framework. They can be used to support the applications intelligently.
- (3) This paper reported the implementation in logistic and proofed the effectiveness.

The rest of this paper is organized as follows: Section 2 gives an introduction to the related works. Section 3 discusses the problems we have met in the transaction control of IoT. Section 4 describes the architecture of the framework and Section 5 describes the design of the top ontologies. In Section 6, we give the logistics scenario in different environments and discuss their workflow. Finally Section 7 gives the conclusion and the future work directions.

2. Relate Work

The current researches on Internet of Things are deployed in three directions: things, networks, and intelligence. In the direction of things, the major research is the frameworks of IoT, which is targeted at the visibility improvement of things, for example, identification and investigation. The most famous architecture is EPCGlobal [4] presented by Auto-ID center of MIT and UID [5] put forward by Ubiquitous ID Center. In the second direction, the research includes wireless sensor networks, privacy protection and security. Numerous studies have been committed to improving the energy efficiency of wireless sensor network capacity, scalability, reliability, and robustness [6, 7], for example, 6LoWPAN [8], a combination of the IP protocol and IEEE 802.15.4, lightweight security architecture of networks and smart identification [9]. In the direction of intelligence, researchers have proposed to use distributed artificial intelligence, rich intelligent application (service) to facilitate the problems of coordinating interaction between tasks of IoT [10].

However, the most important thing for the intelligence of IoT is to establish a suitable architecture, where the functions, data process, and message transfer models can subsequently be designed appropriately to construct and control the transactions. In the past few decades, the research on the semantic web has been dedicated to combining the artificial intelligence and knowledge engineering for representation and processing of data and knowledge. The semantic technology with formal descriptions by machines provides a method to describe the heterogeneous objects, information sharing, and integration issues.

The development of a series of semantic web technologies, such as ontology, semantic annotation, bond information, and semantic web services [11], can be used as the main solution to achieve the sharing of semantic information between IoT entities. Among them, the representative studies include Task Computing based middleware [12], Smart Semantic middleware [13], and Semantic Device bus [14]. In 2009, Scioscia and Ruta have proposed to use the technologies of semantic web with IoT and developed the semantic web of things (SWoT) [15]. The main idea is to use the theory of ontology to make IoT intelligent. The University of Munich and DOCOMO Euro-Labs developed the Perci framework to implement basic mechanisms for seamless interaction between mobile services and tagged objects [16]. In 2011, the Semantic Sensor Network Incubator Group has developed the SSN [17] ontology to describe the sensors and the resources in the sensor network. Which provides a high-level description of the behavior of the sensor devices

themselves, observation and deployment aims to make the sensor networks intelligent [18]. Other researchers, such as De et al. [19], proposed a service modeling for the IoT and the main idea is to represent entities and humans as devices or resources and using web services to organize and reasoning. Besides the above, a standardized protocol XMPP [20] had been proposed for communication of IoT and some methods have been proposed for testing its conformance and performance [21, 22].

3. Problem Statement

So far, the research on intelligent of the sensor network and the IoT mostly focus on how to describe, manage, and monitor the entities (sensors in the sensor network and things in IoT) and how to annotate the data acquired from entities to improve the efficiency of retrieval. For example, SPITFIRE [23] framework aims to semiautomatically annotate the entities with linked open data as well as inference and query of the sensors status. The SSN ontology is developed for the purpose of perception of objects, sensor data, and system deployment. The main idea is still central to the sensor and limited to the semantic annotation of data and the data processing is still manual. Although [19] proposed the idea of entities as services, it still limited in representing data.

However, the real IoT is described as “Everything from tyros to toothbrushes from realty to napkins might soon be in communications range” and “from anytime, anyplace connectivity for anyone, we will now have connectivity for anything” [1]. Therefore, IoT should not be limited to the sensor network. Instead, it should be inclusive of all entities and the entities interact and cooperate with each other to complete a variety of simple or complex tasks. In a sense, the information processing in IoT should be automatic, while people become just bystanders as passive observers. For example, a service provider in the logistics system is only concerned about the object delivery, receipt, and costs. The storage time and place, selection of transport vehicles, transportation path determination, and other issues should be decided by the system. So it requires a mechanism to automatically build a chain of activities according to the user needs. Also, it needs to monitor and regulate the execution of activities.

4. Architecture for the Process of Transactions

In this paper, we propose a framework to process the transaction on IoT based on semantic web technologies as shown in Figure 1. By the support of the ontologies defined on the top of the framework, we can implement the functionalities at various levels.

- (1) The main goal of the entities link layer is information transmission and control drive. It is the intermediate layer between logical and entity layers. This layer is not only to achieve the discovery of the newly added entities, converse all kinds of communication protocol, and operate entities, but also to access the

real-time status of entities. It provides a basis for the construction and implementation of IoT transactions.

- (2) The semantic annotation layer annotates metadata with semantic tags and supports the higher layer. In the proposed framework it is also entrusted with the function of package the entities' information as semantic web services.
- (3) The service registry center as UDDI (Universal Description, Discovery and Integrate) can use the standard SOA to store and manage the IoT services.
- (4) The main function of the transaction construction layer is to construct the activity chains as transaction according to the user needs. Service composition algorithms can be used to achieve this goal by taking into account the context, such as the location of the entity and the scheduling order. Furthermore, it needs to consider whether the conditions exist that there are synergies between transactions.
- (5) The transaction execution layer main functions are to perform transaction management and control, including state information of entities in the activity chain, state perception of subsequent entity in the transaction, the entity dynamic replacement or termination when an entity is unable to work and the scheduling and merger between synergies transactions.

5. Top Ontologies for Transactions on IoT

The foundation of aforementioned framework is a series of compliant common ontologies. It should be able to complete the description of transaction logic and become the top to support the whole architecture. We have proposed the following competency questions (CQ) in accordance with the construction method of the literature [24].

CQ1. For a user needed transaction, what conditions does it perform in and what role does it play?

CQ2. What are the premise, impact, and result?

CQ3. What is the series of activities to constitute a particular transaction?

CQ4. How does the activities construct?

CQ5. For a specific activity, what is its execution condition? What kind of output and effect will it cause?

CQ6. For a specific activity, what role does it play in a transaction?

CQ7. What are the constraints for the execution of transactions and activities?

CQ8. When will the transactions and activities start and finish?

We define the top ontologies for the transactions on IoT following the questions above:

$$O_{IoTtrans} \stackrel{\text{def}}{=} \{O_D, O_G, O_R, O_C, O_E, O_S, O_{ST}, O_A, P, \phi\}. \quad (1)$$

In the definition, we have the following:

O_D : for the domain ontology;

O_G : for the goal ontology;

O_R : for the role ontology;

O_C : for the constraint ontology;

O_E : for the message ontology;

O_S : for the status ontology;

O_{ST} : for the space-time ontology;

O_A : for the activity ontology that transactions mentioned;

P : for the process collection;

ϕ : for the transmission functions.

5.1. Domain Ontology. The entities that interact with each other in a particular domain are the essence of the transactions on IoT. Construction and execution of transactions involve the concepts and relations between concepts of the domain, and it should be the basis for the control. Domain ontology can be a closed area defined by domain experts, such as a simple booking system, or a multiple field based on several closed domains, such as the modern manufacturing involves logistics, materials and energy. If the domain is a multiple field, it should combine domains as a closed area which is the significance of IoT to involve everything.

5.2. Goal Ontology. Goal ontology indicates the goals of transactions or activities. It can be explained as a result status of entity that operated by a series of operations. We use it to define the termination of transactions or activities. The goal ontology can be defined as a triple:

$$O_G \stackrel{\text{def}}{=} \{\text{GoalC}, \text{GoalR}, \text{rel}\} \quad (2)$$

GoalC = profile \cup Target \cup Effect \cup Quality is the concept set of goal ontology.

profile is a document that contain the knowledge of goal and its explanation.

Target is the set of operation targets, and can be represented by part of domain ontology. It involves the entities affected by the transaction.

Effect means the effect result. It is represented by entities' status. The entity changes into the expected status when the goal has been achieved.

Quality refers to the factors of quality, such as quality, cost, and reliability.

GoalR is the relationship between the concepts, for example, Target has Effect.

rel : GoalC \times GoalC \rightarrow GoalR is a mapping on the goal ontology.

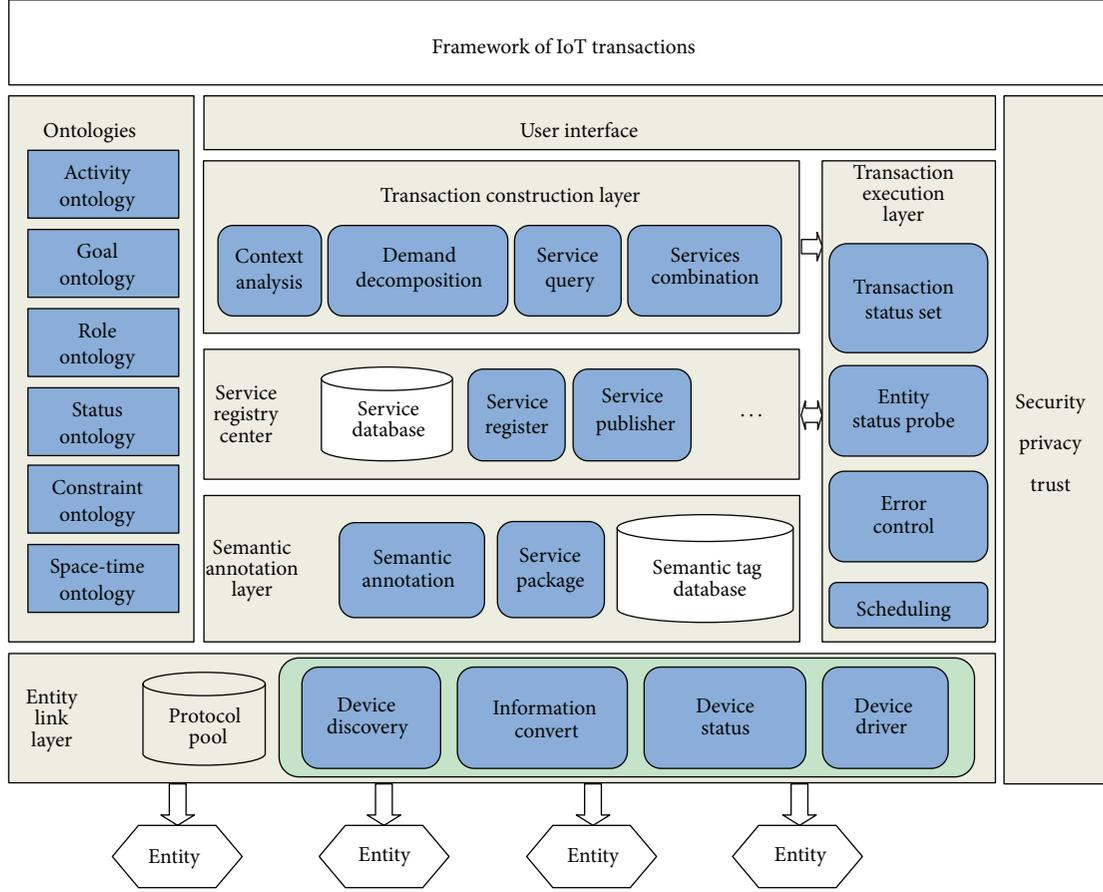


FIGURE 1: Framework for the transactions on IoT.

5.3. *Role Ontology*. The main idea of role ontology is to define roles for the entities in transactions and how they should perform. For example, trucks act as conveyance in transport but act as operation targets in refuel event. Meanwhile, the definition of role ontology provides means for security, trust, and privacy. The role ontology can be defined as a quadruple:

$$O_R \stackrel{\text{def}}{=} \{\text{RoleC}, \text{RoleR}, \text{rel}, \text{RoleA}_0\} \quad (3)$$

RoleC = profile \cup Actor \cup Type is the concept set.

profile is a document that contains the knowledge of role and its explanation.

Actor is defined by roles in the domain ontology to show the real role each entity has played in the transaction, such as manager and employee.

Type is the type of actor; generally, it is requester or service provider or resource owner or operator.

RoleR is the relationships between roles; for example, employee managed by manager.

rel : RoleC \times RoleC \rightarrow RoleR is a mapping on the role ontology.

RoleA₀ means the axiom in the role ontology, used to derive implicit relationships between concepts and the relationship constraints.

5.4. *Constraint Ontology*. Constraint ontology is defined as data constraint of entities. It indicates the next events trigger as well as end conditions or decides which direction the transaction should be going. The constraint ontology not only guides the construction of transaction as QoS, but also judges the status and environment to make decisions. The ontology can be defined as a triple:

$$O_C \stackrel{\text{def}}{=} \{\text{ConstraintC}, \text{Logical Expressions}, R^N\} \quad (4)$$

ConstraintC = profile \cup Attribute \cup Range is the concept set.

profile is a document that includes the ID and the description information.

Attribute refers to the attribute constraint, represented by entities' properties and functions. It is a subset of domain ontology.

Range is the range of Attribute.

Logical Expressions = $\langle \text{Attribute, Range, } f_R, (0, 1) \rangle$ is defined as a four-dimensional Cartesian product. Attribute and Range are defined as above. f_R is a set of relation functions, and $(0, 1)$ is a Boolean set. The Logical Expressions can be expressed as follows:

$$f_R(\text{Attribute, Range}) \longrightarrow (0, 1). \quad (5)$$

R^N is a n -ary operation on the Logic Expressions to complex the logical expressions.

5.5. Message Ontology. The trigger message for the subsequent activity and the information exchange between the parallel activities can be represented by message ontology. We define the message ontology as a quadruple:

$$O_E = \{\text{Sender, Receiver, Type, Content}\}. \quad (6)$$

Sender and Receiver denote the sender and receiver of messages.

Type refers to the message type, for example, data message, control message, begin message and end message, and so forth.

Content is the content of messages.

5.6. Status Ontology. It is noted that the status ontology is used to indicate the state of entities in transactions, including materials, products, parts, personnel, tools, data, and other information about the status. The information represented by status ontology not only participates in the construction of transactions as determined constraints, but also dynamically is updated during the execution of a transaction and available to determine the constraint conditions of the next step. The status ontology can be defined as a two-tuple:

$$O_S = \{\text{Status, Value}\},$$

$$\text{Status} = \text{Attribute} \cup \text{Resources} \cup \text{Products} \quad (7)$$

$$\cup \text{Opertors} \cup \text{Tools} \cup \text{Environment}.$$

The definition of Attribute is the same as the definition in Section 5.4.

Resources, Products, Opertors, and Tools denote materials, products, operators, and tools just as literally the same. They can be defined by domain ontology and particular they will be involved to different domain ontology.

Environment is used to describe the state of the environment in which the entities may perform different functions in different environment and it is much important for the control.

Value is the range of Status.

What needs to be addressed is that the status is dynamic for the transactions on IoT. The statuses focused are those changes with time, location and environment, and so forth and affect the activities or transactions. The static properties, commonly, are not consisted.

5.7. Space-Time Ontology. Space-time ontology is used to represent temporal and spatial properties of entities or activities. For example, in the field of GIS, the ontology has been carried on the thorough research [25, 26]. The spatial properties of entities restrict the construction of transaction generally. Both of spatial and temporal properties restrict the entities or activities when the transactions are executing. It must be in consideration that spatial and temporal properties are one of the most important factors for synergy. The space-time ontology in our opinion is an abstract ontology and it has two subontologies: space ontology and time ontology. We use our method for two reasons, even if the two functions are all works. Firstly, the entities in our approach may be real items or data; we should consider the time and location for the real items but only focus on the attribute of time and ignore the location attribute for data; so we only need the subontology of time ontology. Secondly, for the implementation using java language, it does not support multiple inheritances and must use interfaces which increases the difficulties of development. We simply defined the space-time ontology in Figure 2.

5.8. Activity Ontology. Activity is the most basic unit of the transaction process. The activity ontology describes the concepts, functions, and situations of each act or step. In the transactions on IoT, activity means not only the data process but also the functions of entities represented as web service. The activity ontology defined as follows:

$$O_A = \{\text{Input, Output, trigger message, exception, status, participant, conditions}\} \quad (8)$$

Input, Output may be data or entities defined by domain ontology.

trigger message $\subseteq O_E$ is a message set for trigger activities.

exception explains the exceptions in the activity and commonly generates a message to the transaction execution layer.

status is only defined on entities and refers to the status of them, same defined as Section 5.6.

participant $\subseteq O_R$ is used to identify the role of activities.

conditions = {precondition, postcondition} indicates the activity state precondition and postcondition.

5.9. Process Ontology. The process ontology is used to describe the process sequence relationships between activities. We direct reference to the definition in OWL including $\langle \text{Sequence} \rangle$, $\langle \text{If-Then-Else} \rangle$, $\langle \text{Choice} \rangle$, $\langle \text{Any-Order} \rangle$, $\langle \text{Split} \rangle$, $\langle \text{Split-Join} \rangle$, $\langle \text{Repeat-While} \rangle$, and $\langle \text{Repeat-until} \rangle$ [27].

5.10. Relations between the Top Ontologies. The relationships between the top ontologies show in Figure 3. We created the top ontologies by Protege4.1 and the part of codes showed in Algorithm 1.

```

<owl:Class rdf:about="&IoTrans;ActivityOnto">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  .....
</owl:Class>

<owl:Class rdf:about="&IoTrans;ConstraintOnto">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  .....
</owl:Class>

<owl:Class rdf:about="&IoTrans;DomainOnto">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  .....
</owl:Class>

<owl:Class rdf:about="&IoTrans;GoalOnto">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  .....
</owl:Class>

<owl:Class rdf:about="&IoTrans;MessageOnto">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  .....
</owl:Class>
.....

<owl:ObjectProperty rdf:about="&IoTrans;Trigger">
<rdfs:range rdf:resource="&IoTrans;ActivityOnto"/>
<rdfs:domain
rdf:resource="&IoTrans;MessageOnto"/>
</owl:ObjectProperty>

<owl:ObjectProperty
rdf:about="&IoTrans;expressedBy">
<rdfs:domain
rdf:resource="&IoTrans;ConstraintOnto"/>
<rdfs:range
rdf:resource="&IoTrans;Space-timeOnto"/>
<rdfs:range rdf:resource="&IoTrans;StatusOnto"/>
<rdfs:subPropertyOf
rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<owl:ObjectProperty
rdf:about="&IoTrans;hasConstraint">
<rdfs:domain rdf:resource="&IoTrans;ActivityOnto"/>
<rdfs:range
rdf:resource="&IoTrans;ConstraintOnto"/>
<rdfs:domain rdf:resource="&IoTrans;Transaction"/>
</owl:ObjectProperty>
.....

```

ALGORITHM 1: The part of codes of top ontologies and their relationships.

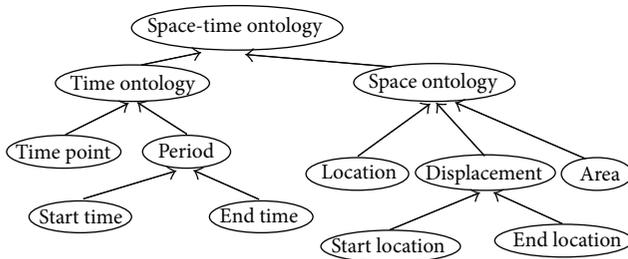


FIGURE 2: Space-time ontology.

6. Case Study of Logistics

In this section we take the third-party transport logistics as a case study to evaluate the proposed framework. In this scenario we compare our approach with the traditional transport logistics and the transport logistics using web services.

6.1. The Traditional Transport Logistics. Figure 4 shows the traditional business process of the third-party logistics. The dotted line indicates the internal information flow and the solid line indicates the information exchange between businesses. This business process includes the following steps. (1) The customer sends a business application to the known logistics enterprises. (2) The manager queries the internal database and confirms whether it can meet the customer's requirements. (3) The manager confirms the order with the customer and arranges the corresponding logistics services.

(4) The customer hands over the goods to the logistics company. (5) The logistics company transports goods to the destination. (6) After the arrival of the goods, employees report the progress to the manager. (7) The manager informs the customer of arrival and accomplishes the rest of the interaction process.

It should be noted that in the traditional third-party logistics enterprise for transportation equipment, such as vehicles, the management is carried out by specialized employees and input the information into the enterprise database manually. It can be seen that the workflow of traditional third-party logistics is still a manual-based and thus less efficient.

6.2. The Transport Logistics Using Web Services. With the development of IoT, more and more relevant technologies are getting used in the logistics industry, such as decision system [28], which has greatly improved the efficiency of logistics management. The most effective method is introduction of web services to logistics systems. Logistics services have been packaged into web services and published in the registry center, which has improved the efficiency of information dissemination and retrieval [29, 30]. With this technology the third-party logistics transport workflow is depicted in Figure 5.

In this scenario, the business process has been changed into the following steps: (1) the logistics enterprises register their services in the registry center and (2) the customers query in the services center to find the services that meet their requirements. (3) After the customers get the needed

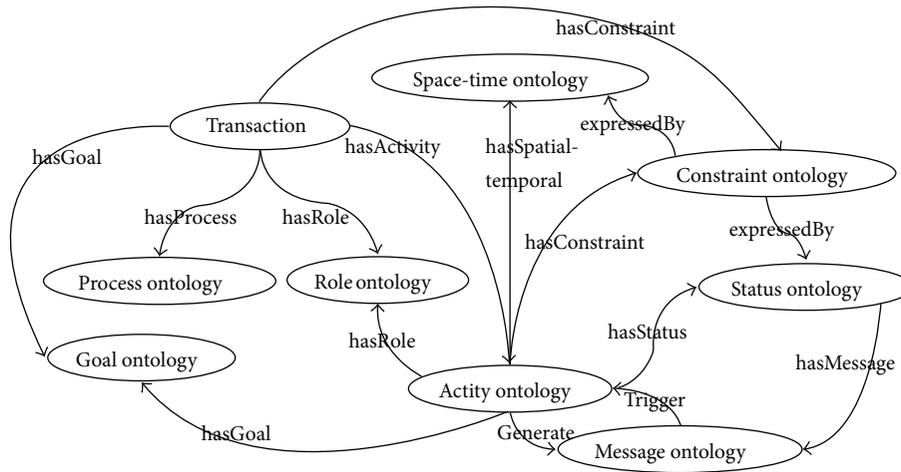


FIGURE 3: Relationships between top ontologies for transactions on IoT.

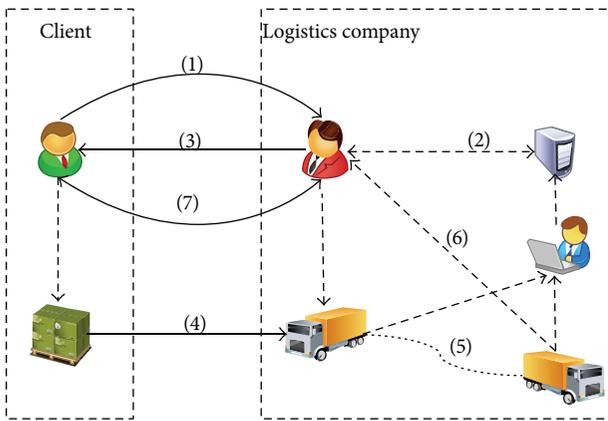


FIGURE 4: The traditional business process of the third-party logistics.

information, they can make their application from user interface. (4) The manager arranges for the corresponding services after receiving the request through the internal management system. (5) Customer hands over the goods to the logistics company. (6) The logistics company transports goods to the destination. (7) After the arrival of the goods, employees report the progress to the manager. (8) The manager informs the customer of arrival and accomplishes the rest of the interaction process.

From Figure 5, we can see that the use of web services has automated the information process. For instance, customers can get the suitable services provided by logistics enterprises without knowing the contact with the support of web service discovery and composition algorithms which reduces the human interaction. However, in this scenario, the management of the equipment still needs through specialized staff. The services registered are static services and cannot schedule the logistics entities automatically (see step (4)).

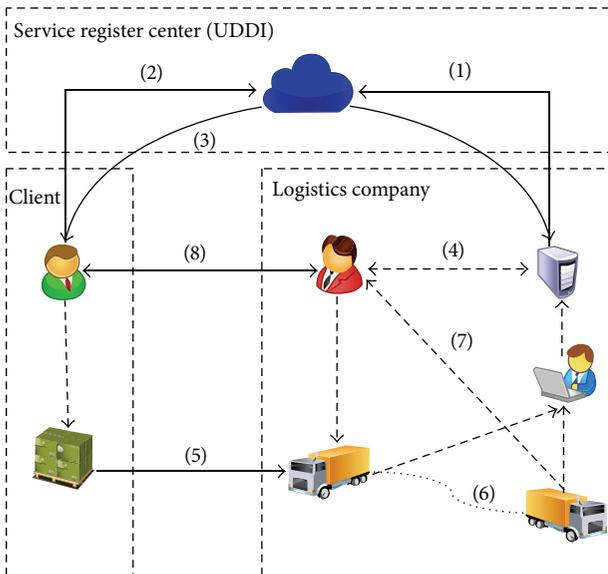


FIGURE 5: The third-party logistics transport using web services.

6.3. The Transport Logistics Using Our Framework. See Figures 6 and 7.

6.3.1. Ontology Design. As shown in Figure 6, based on the top ontologies, we have designed the truck ontology and individuals by Protege4.1.

After analysis, trucks should have different roles and activities during the whole transport. So we used the top ontologies to define the truck ontology dynamically, as in Figure 7.

It can be seen that in Figure 7 the truck plays a role of “Container” when it has activities of “Load” (a) and “Discharge” (c) and plays a role of “Transporter” (b) when it has an activity of “Transport.” Other properties, such as “hasSpatial-Temporal” and “hasStatus,” are defined in Figure 7 too, where we omitted some data properties for the sake of clarity.

The next thing we care about is “Activities.” The three activities, for example, are “Load,” “Transport,” and “Discharge”; we defined them in Figure 8.

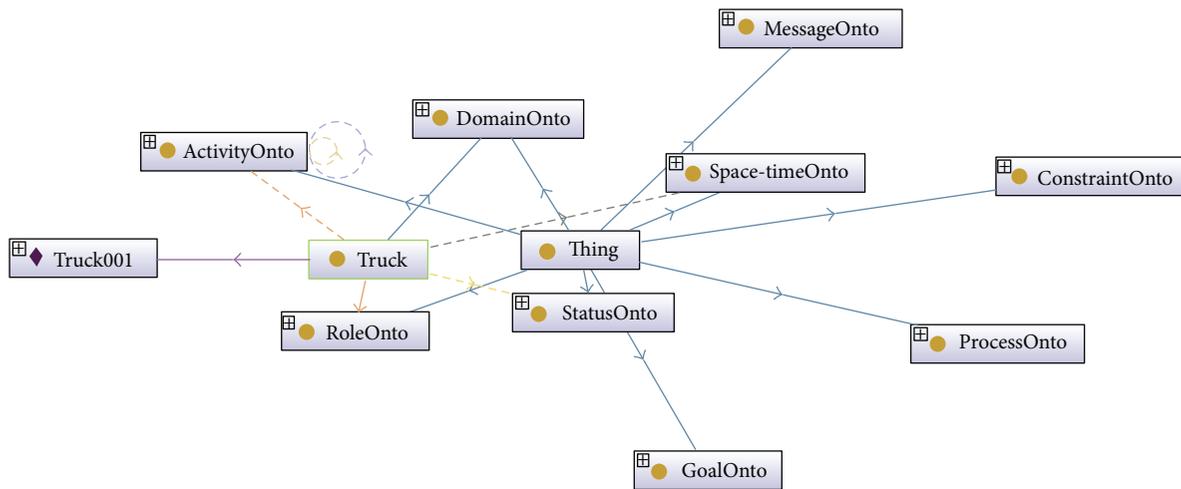


FIGURE 6: Truck ontology and relationships.

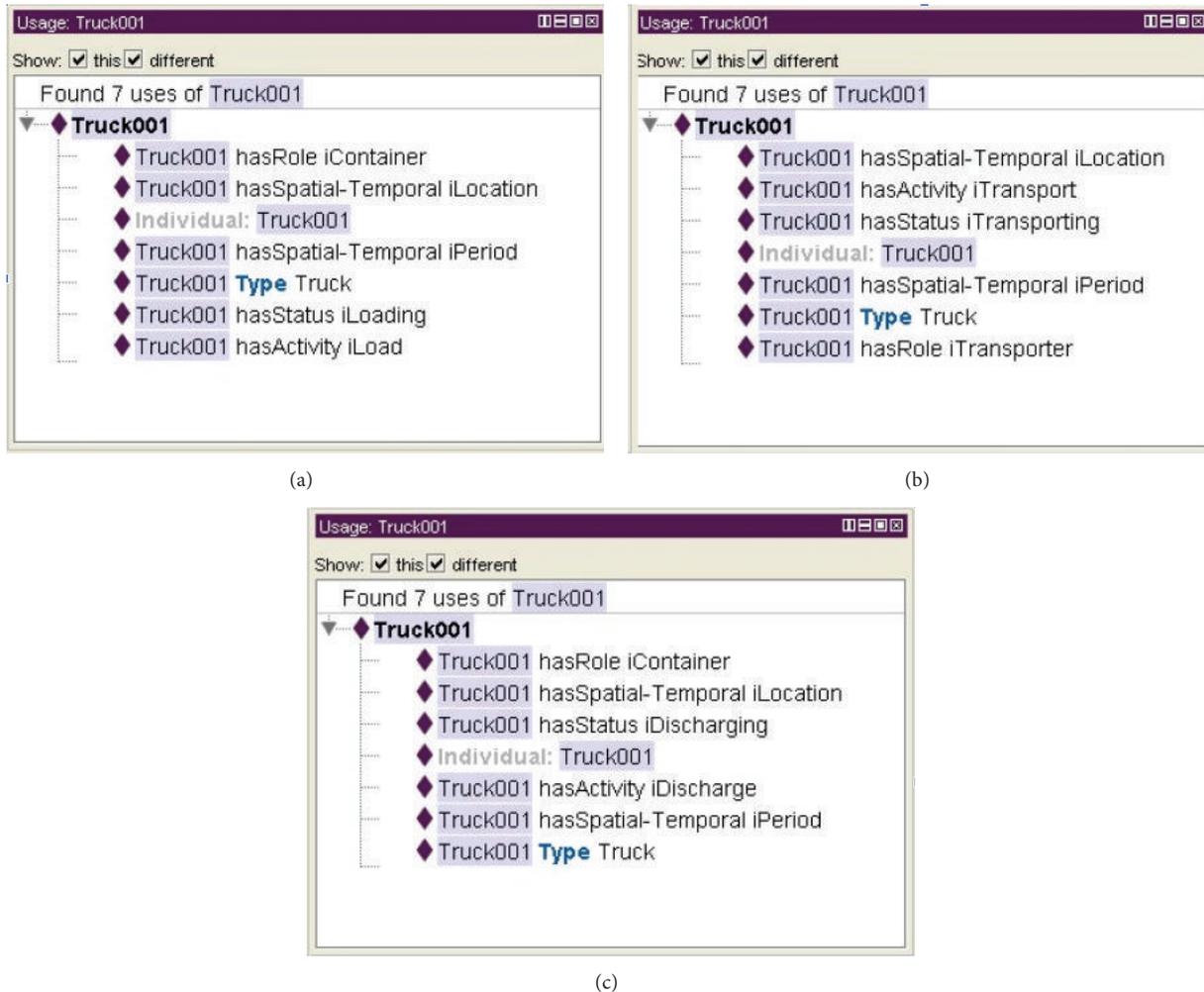


FIGURE 7: Dynamic definitions of truck ontology.

To answer the CQs in Section 5, we can get the following information from Figures 7 and 8: “Load” is an activity of “truck001” (Figure 8(a)) and its goal is “ToLoad,” “truck001” plays a role of “Container” (Figure 7(a)), when the activity starts at some “Location” under constraint of “Load_Cons,” and the entity “truck001” will “hasStatus loading” for a “period” and the activity will continue for a “period.” After loading it will create a “LoadedMessage” to the next activity as a “Trigger.” Moreover, we can see that the next Activity is “Transport” (Figure 8(b)) and the order is “Sequence.” The next role (Figure 7(b)) and activity (Figure 8(b)) continued automatically and so on.

6.3.2. Use Case of Transport Logistics. In our framework, with the top ontologies, each entity is described as a dynamically structured web service that changes with the status and the context of the entity. As shown in Figure 9, with transport entities like trucks as services, the whole business process now runs by interacting these services.

In this scenario, the business process now consists of the following steps: (1) the logistic enterprises register their transport entities as web services in the registry center. (2) Customers query the service to find the entities to meet their requirements and (3) after getting the suitable services information, the customer applies to the entities of the logistics enterprise through the transaction execution layer (under the support of security and privacy). (4) Customer hands over the goods to the logistics company. (5) The logistics company transports goods to the destination and sends an arrival message to the customer automatically by the sensor. (6) The manager accomplishes the rest of the interaction process.

Particularly, the following should be noted.

- (i) The services registered in addition to the enterprises also include the entities of enterprises and the functions of entities are registered as activities. The entity activities bond with enterprises with roles.
- (ii) The transaction construction layer decomposes the requirement and mapped to some composition acts according to specific constraints steeply. There is a difference with forward situation as the requirement may be met by several services of different companies.
- (iii) The entity’s property (such as time, space, and status) will change timely and the change will be captured by sensors or RFID. The data will be sent to the enterprise information system as messages by wireless network. When it happens, the enterprise information system send a message to service registry center and the entity services will be changed (as dotted lines “a” in Figure 9).
- (iv) Whenever the goods load or transport both of the customer and the logistics enterprise can obtain the transport status by querying the services (showed in Figure 9 dotted lines “b”).

The framework greatly reduces the intervention of human beings and in some special conditions the customer can be an

entity, too. The framework supported by the top ontologies realizes direct communication between things and automatic generation and control of transactions.

6.4. Evaluations and Discussion. The main idea of our approach is to automatically construct and execute the transactions on IoT to decrease the communications between entities and human beings. For a logistics company it should decrease the operating costs and increase the efficiency and earnings.

Obviously, due to e-commerce model (the transport services published in the Internet and clients can use e-business platform to fulfill transactions) and cloud technology, our approach and the related methods of web services have an advantage over the traditional transport logistics. As mentioned in [31], the logistics industry had been and will be keeping on growing exponentially. The investment in advance for e-commerce will be recovered soon and the earnings will increase rapidly.

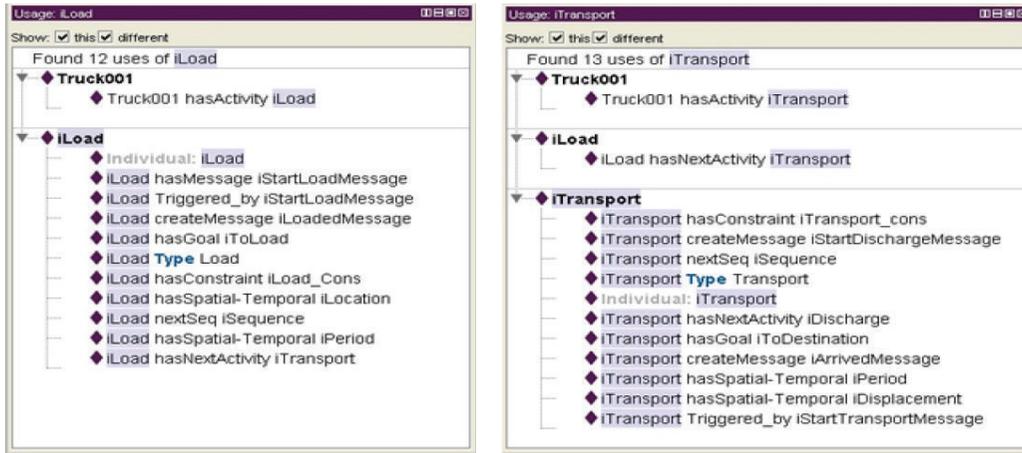
The biggest difference between Figure 5 and Figure 9 is that in the scenario of Figure 5 the company processes the internal data manually but it is automatic in the scenario of Figure 9. Suppose the labor cost per transaction is α and the growth rate is k in Figure 5. The investment in advance (for sensors and network) is β and the maintenance cost per transaction is ε in Figure 9. Besides the necessary costs, the extra costs of each transaction show in Figure 10 and the blue solid lines denote the function of $y = (1 + k)\alpha$ which is the costs of each transaction in Figure 5 and the red dotted lines denote the function of $y = (\beta/x) + \varepsilon$ which is the costs of each transaction in Figure 9. When the amount of the transactions is more than θ the average extra costs in Figure 9 will be lower than in Figure 5. That means our approach performs better economically.

From efficiency point of view, the whole process is automated and the manager only acts the last actor in Figure 9 (action (6)) that is automated. But in Figure 5, after clients make an order (action (3)), the manager should inquire and schedule (action (4)) that is semiautomated. Due to the reduction of certain aspects such that the company manager arranges for the corresponding services and communicates with clients and internal schedule, our approach is better than the method of using web services in terms of efficiency.

Besides all the advantages above, our approach will decrease the error rate of schedule and provide a better user experience, for clients can keep abreast of the logistics situation through the cloud platform using the date that the sensors generated.

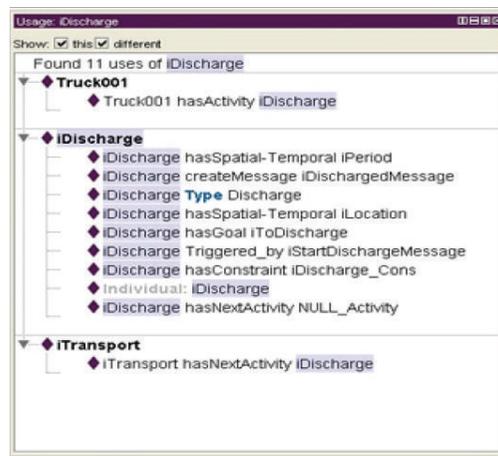
7. Conclusion

In order to achieve the goal of intelligent IoT, we have proposed a framework for the construction and execution of the transactions on IoT. To support the framework from semantic layer we have designed the top ontologies. The main idea is to use semantic technology to describe the entities as dynamic web services, construct transactions with web



(a)

(b)



(c)

FIGURE 8: Definition of activities of “Load,” “Transport,” and “Discharge.”

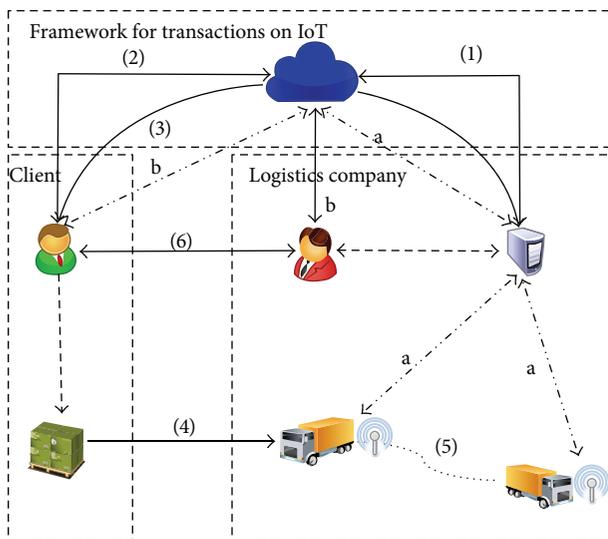


FIGURE 9: The transport of the third-party logistics under the framework of transactions on IoT.

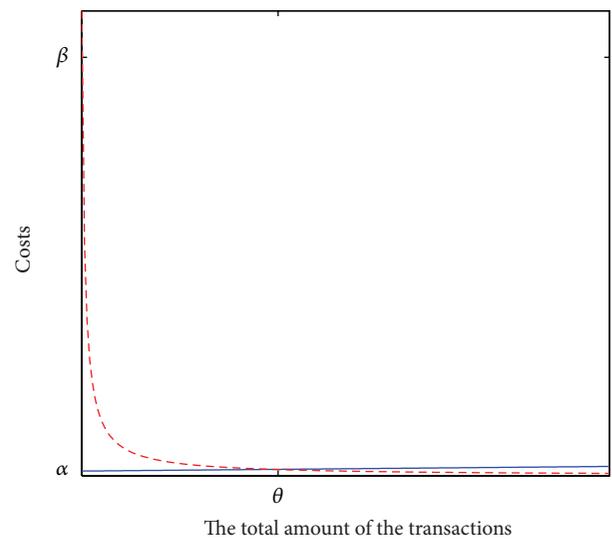


FIGURE 10: Costs of each transaction in Figures 5 and 9.

service discovery and composition technology, and control their executions.

Our framework provided an approach to the intelligence of IoT. In our architecture we can package the entities as web services and using semantic web technology to reason about them. The designed top ontologies can be a guide of how to use our architecture.

The framework has worked well with the support of the top ontologies in the field of logistics and thus minimized the intervention of people. However, there are still some problems that remain open.

- (i) The web services described by OWL-S are not enough to depict the dynamic services; the provided components (ServiceProfile, ServiceModel, and ServiceGrounding) need to be extended because of having no abilities to describe dynamic status.
- (ii) Algorithms of service discovery and composition can be used to construct transactions, but it does not consider the status and environment as the factor. So the new algorithms need to be designed to address these issues.
- (iii) The most special characteristics of transactions on IoT are the correlation of entities, which is different from that in the information space. When errors occur in information space, it can be eliminated by using reset or rollback. In the case transactions on IoT, however, we need another treatment for the error.
- (iv) We are still in the research of how to implement the combination and synergy to the transactions. It needs more complex algorithms and schedule supports.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the 3th strategic rising industry program of Guangdong Province, (Project no. 2012556003); International Cooperation Special Program for platform, (Project no. 2012J510018); the key lab of cloud computing and big data in Guangzhou (Project no. SITGZ[2013]268-6); Engineering & Technology Research Center of Guangdong Province for Big Data Intelligent Processing (Project no. GDDST[2013]1589); Engineering & Technology Research Center of Guangdong Province for Big Data Intelligent Processing (Project no. GDDST[2013]1513-1-11).

References

- [1] I. Peña-López, "ITU Internet report 2005: the internet of things," Tech. Rep., 2005.
- [2] F. Scioscia and M. Ruta, "Building a semantic web of things: issues and perspectives in information compression," in *Proceedings of the IEEE International Conference on Semantic Computing (ICSC '09)*, pp. 589–594, Berkeley, Calif, USA, September 2009.
- [3] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.
- [4] K. Traub, G. Allgair, H. Barthel et al., "The EPC global architecture framework," EPC global Ratified specification, 2005.
- [5] Ubiquitous ID Center, "Ubiquitous ID architecture," T-Engine Forum Ubiquitous ID Center Document, Ubiquitous ID Center, 2006.
- [6] A. Dunkels and J. P. Vasseur, "IP for smart objects," IPSO Alliance White Paper 1, 2008.
- [7] A. C. Sarma and J. Girão, "Identities in the future internet of things," *Wireless Personal Communications*, vol. 49, no. 3, pp. 353–363, 2009.
- [8] J. Hui, D. Culler, and S. Chakrabarti, "6LoWPAN: incorporating IEEE 802.15. 4 into the IP architecture," IPSO Alliance White Paper 3, 2009.
- [9] C. Wang, G.-Y. Hu, and H.-G. Zhang, "Lightweight security architecture design for wireless sensor network," *Journal of China Institute of Communications*, vol. 33, no. 2, pp. 30–35, 2012.
- [10] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [11] A. Gómez-Pérez and R. Benjamins, "Overview of knowledge sharing and reuse components: ontologies and problem-solving methods," in *Proceedings of the IJCAI and the Scandinavian AI Societies, CEUR Workshop*, 1999.
- [12] Z. Song, A. A. Cárdenas, and R. Masuoka, "Semantic middleware for the internet of things," in *Proceedings of the 2nd International Internet of Things Conference (IoT '10)*, pp. 1–8, Tokyo, Japan, December 2010.
- [13] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan, "Smart semantic middleware for the internet of things," in *Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics (ICINCO '08)*, pp. 169–178, Funchal, Portugal, May 2008.
- [14] K. Yang, S. Li, L. Zhang, and G. Pan, "Semantic device bus for internet of things," in *Proceedings of the 8th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC '10)*, pp. 341–346, Hong Kong, China, December 2010.
- [15] F. Scioscia and M. Ruta, "Building a Semantic Web of Things: issues and perspectives in information compression," in *Proceedings of the IEEE International Conference on Semantic Computing (ICSC '09)*, pp. 589–594, Berkeley, Calif, USA, September 2009.
- [16] G. Broil, M. Paolucci, M. Wagner, E. Rukzio, A. Schmidt, and H. Hußmann, "Perci: pervasive service interaction with the internet of things," *IEEE Internet Computing*, vol. 13, no. 6, pp. 74–81, 2009.
- [17] M. Compton, P. Barnaghi, L. Bermudez et al., "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, 2012.
- [18] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the internet of things: early progress and back to the future," *International Journal on Semantic Web and Information Systems*, vol. 8, no. 1, pp. 1–21, 2012.
- [19] S. De, P. Barnaghi, M. Bauer, and S. Meissner, "Service modelling for the Internet of Things," in *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS '11)*, pp. 949–955, IEEE, September 2011.
- [20] P. Saint-Andre, "Extensible messaging and presence protocol (XMPP): core," RFC 6120, Internet Engineering Task Force (IETF), 2011.

- [21] R. Klauck and M. Kirsche, "Chatty things - Making the Internet of Things readily usable for the masses with XMPP" in *Proceedings of the 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom '12)*, pp. 60–69, October 2012.
- [22] X. Che and M. Stephane, "A passive testing approach for protocols in internet of things," in *Green Computing and Communications (GreenCom), IEEE and Internet of Things (iThings/CPSCoM), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pp. 678–684, 2013.
- [23] D. Pfisterer, K. Römer, D. Bimschas et al., "SPITFIRE: toward a semantic web of things," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 40–48, 2011.
- [24] N. F. Noy, "Ontology development 101: a guide to creating your first ontology: knowledge systems laboratory," Tech. Rep. KSL-01-05, SMI-2001-0880, Stanford University. Stanford Knowledge Systems Laboratory, Stanford Medical Informatics, Stanford, Calif, USA, 2001.
- [25] B. Vasseur, D. van de Vlag, A. Stein, R. Jeansoulin, and A. Dilo, "Spatio-temporal ontology for defining the quality of an application," in *Proceedings of the International Symposium on Spatial Data Quality*, pp. 67–81, Bruck an der Leitha, Austria, 2004.
- [26] B. A. El-Geresy, A. I. Abdelmot, and C. B. Jones, "Spatio-temporal geographic information systems: a causal perspective," in *Advances in Databases and Information Systems*, vol. 2435, pp. 191–203, Springer, Berlin, Germany, 2002.
- [27] D. Martin, M. Burstein, J. Hobbs et al., "Owl-s: semantic markup for web services," 2004, <http://www.ai.sri.com/daml/services/owl-s/1.2>.
- [28] Y. Zhu and S. Zhang, "Research and analysis on the construction of e-commerce intelligent decision system of logistics enterprise," in *Proceedings of the 2nd International Conference on Science and Social Research (ICSSR '13)*, 2013.
- [29] A. Talevski, E. Chang, and T. S. Dillon, "Reconfigurable web service integration in the extended logistics enterprise," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 2, pp. 74–84, 2005.
- [30] C. Preist, J. Esplugas-Cuadrado, S. A. Battle, S. Grimm, and S. K. Williams, "Automated business-to-business integration of a logistics supply chain using semantic web services technology," in *The Semantic Web—ISWC 2005*, vol. 3729 of *Lecture Notes in Computer Science*, pp. 987–1001, Springer, Berlin, Germany, 2005.
- [31] China Competition Information Center, *China E-Commerce Logistics Industry Report (2014–2018), Developing Mode and Investment Strategy Planning*, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

