

# Constructing the Web of Events from raw data in the Web of Things

Yunchuan Sun<sup>a</sup>, Hongli Yan<sup>a</sup>, Cheng Lu<sup>b</sup>, Rongfang Bie<sup>b,\*</sup> and Zhangbing Zhou<sup>c,d</sup>

<sup>a</sup>*Business School, Beijing Normal University, Beijing, China*

<sup>b</sup>*College of Information Science and Technology, Beijing Normal University, Beijing, China*

<sup>c</sup>*School of Information Engineering, China University of Geosciences, Beijing, China*

<sup>d</sup>*Computer Science Department, Institute Mines-TELECOM/TELECOM SudParis, Paris, France*

**Abstract.** An exciting paradise of data is emerging into our daily life along with the development of the Web of Things. Nowadays, volumes of heterogeneous raw data are continuously generated and captured by trillions of smart devices like sensors, smart controls, readers and other monitoring devices, while various events occur in the physical world. It is hard for users including people and smart things to master valuable information hidden in the massive data, which is more useful and understandable than raw data for users to get the crucial points for problems-solving. Thus, how to automatically and actively extract the knowledge of events and their internal links from the big data is one key challenge for the future Web of Things. This paper proposes an effective approach to extract events and their internal links from large scale data leveraging predefined event schemas in the Web of Things, which starts with grasping the critical data for useful events by filtering data with well-defined event types in the schema. A case study in the context of smart campus is presented to show the application of proposed approach for the extraction of events and their internal semantic links.

Keywords: Web of Events, Web of Things, restful, information extraction, mobile

## 1. Introduction

The Web of Things (WoT) aims at enhancing our daily life through deriving knowledge such as events (to be) happened, from the trajectories of things and people which reflect activities across the cyber world and the physical world. WoT envisions a brand-new Web that promises to provide a platform and a friendly interface for people and smart things to access the status of physical objects, where Representational State Transfer (REST) can be viewed as an architecture to support WoT [1]. Generally, WoT is enabled to sense, capture and process more data with the development of related technologies such as wireless sensor network, RFID technology and semantic technologies. Massive data are continuously generated by trillions of sensing devices in the physical world. It is impossible for people to understand and use these raw data in problem solving due to the characteristics of data such as massiveness, complexity, trivialness, dynamic and implicit semantics, which means it is rather hard for people and smart things to master valuable information hidden in the massive data.

Firstly, people are more inclined to visit intuitive, easy-to-understand and easy-to-use integrated event information rather than complex raw data. For example, it is difficult for people to understand the

---

\*Corresponding author: Rongfang Bie, College of Information Science and Technology, Beijing Normal University, Beijing, China. E-mail: rfbie@bnu.edu.cn.

meaning of a set of data like (objectID: 867495012228536, TimeStamp: 2012-03-2620:41:31, Location: (latitude=39.9661140441895, longitude=116.372123718262)) reported by the mobile device of a student, while event information like “a student has a class at 20:41:31 2012-03-26 for 53 minutes at the place of Classroom 8” is more useful and understandable for people.

Secondly, the internal semantic links among different events are more helpful and meaningful than isolated activities for people to master the situations and to solve problems. For example, a semantic link like “an event *Being absent* caused by *Playing sports*” is more useful than two isolated events “*Being absent*” and “*Playing sports*”. Links between events weave the isolated events into a connected network. By using the semantic links among events, we can find out more inherent and implicit relations between events by reasoning among the existed links. Such a network of events can provide a global view of the situation for users by revealing the internal relationships between events. More useful evolving patterns can be mined from the network which can be used to predict future trends in the field. For example, we can find the inherent reasons of some absence event by the transitivity of the *causeOf* links among a series of events.

Nowadays, more and more researchers realize the importance of events to the web, and believe that events are the fundamental abstractions in studying the dynamic world. Besides, there are gradually a few works focusing on events in the web. For example [2], proposes that the evolving dynamic web would be a ‘web’ of interconnected events across time and space, and the paper will mainly focus on interconnections between events as they occur across space and time. Based on Twitter micro blogs, [3] studies the structural analysis of the emerging event-web; what the authors do is using occurrence and co-occurrence patterns to study Power Law in the web, and they also intend to study causality across such events in future research. Therefore, how to process these raw data efficiently and how to extract valuable knowledge from these data play key roles as well as key challenges in the development of WoT.

When extracting events in the domestic environment, the concept of event and event types should be defined, which will be further discussed in Section 3. Generally, there might be some crucial thresholds to identify an event of a certain type. For example, an event of *Having class* occurs for a student only when he is in the classroom for more than 45 minutes. In many cases, events can be identified efficiently by using a set of predefined event thresholds. Similarly, semantic links among events can also be discovered according to a list of filters on the properties of the events. Based on these ideas, an event schema which is a list of definitions on different kinds of events and different kinds of links can be worked out by domain experts; then the events and their links can be extracted by using the event schema and a network of events can be constructed; and finally we can present the events on the Web in a visualized way. Therefore, in Section 3, we introduce the concepts of event, event type, link type and event schema in WoT. Section 4 discusses the model for extracting event information which includes three layers, i.e. data collecting, event information extracting and event presenting.

At last, a case study in the smart campus has been implemented in Section 5 showing how the proposed model can be used in practice, in which our model of Event Information Extracting adopts a process procedure similar to the ECA rules (*Event-Condition-Action*). ECA rules have been widely used in various researches including semantic web. For example, [4] creates Event-Condition-Action Rule Languages in semantic web. In our model, the emergence of real data in data collecting layer (as *Event* in the ECA rules) will trigger the process of event extracting layer, where the agent judges whether the precondition (as *Condition* in the ECA rules) is satisfied; and then the critical and sensitive data meeting the conditions of the event types would be utilized to generate events (as *Action* in the ECA rules). Obviously, there is difference between our prototype and the ECA rules, for the concept of event in ECA rules differentiates from the events we want to extract by using our model.

The contributions of this paper include (i) propose an efficient approach to extract event information, especially internal semantic links between events from detailed and large scale data based on predefined event schema in WoT, and (ii) use a visualized way to present the acquired events and their internal links on WoT and to facilitate the user accessing the system.

## 2. Related work

Many efforts have been made in the fields of data management and information processing for achieving the vision of the Web/Internet of Things. Actually, researchers have done a lot of work in three different aspects of this area. A long list of challenges such as data storage, data preprocessing like cleaning and filtering, consistency checking and semantic explaining have been discussed in [5–8]. Database management system is a good start for managing sensor data. But existing database systems are somehow not suitable to be used directly for handling live sensor data [9–11]. To reduce the cost of communication, infrastructure redesign is required for data management through fully exploiting the local storage and processing capability, and a proposed architecture for managing data, i.e., *StonesDB*, is utilized for storing and processing data in sensor networks [12]. Regarding the data storage in wireless sensor networks, two approaches named as *ODS* (optimal data storage) and *NDS* (near-optimal data storage) are proposed by formalizing the storage problem into either a one-one (one producer and one consumer) or a many-many (m producers and n consumers) model [13]. Several principles are proposed for *RFID* data management, where the primary purpose is to digest the data close to the source [14]. A distributed *RFID* system is presented for efficiently cleaning, filtering and augmenting the raw data generated by tag readers [5]. Dynamic Relationship *ER* Data Model, as well as a corresponding system – Siemens *RFID* Middleware – is proposed to manage the general temporal-oriented data. Besides, this model supports *RFID* data tracking and monitoring [15]. In addition to the efforts of dealing with data from *WSNs* or *RFID* readers, some works focus on heterogeneous data from different data sources. *INDAMS* is an integrated network and data management system used for managing data in heterogeneous *WSNs* [16], while a web service based schema also aims at managing heterogeneous data in the Internet of Things [17]. A scalable lean data provision architecture based on ontology is developed to deal with the massive data in a semantic way [18]. Most of these works focus on how to store and manage the captured data rather than how to explain or extract meaningful event information.

Some works focus on the framework of data presentation and analysis. An architecture named *Sem-Sense* is proposed for collecting real world data from a physical system of sensors and publishing the data on the Web [19]. In order to find potential and useful spatio-temporal patterns from the sensor data, a data model *STSG* (spatio-temporal sensor graphs) has been developed [20]. Those works lay emphasis on the presentation and visualization of the data but not the information extracting.

Moreover, there have been also several researches on activity recognition from the sensor data [21–32]. There are many works focusing on activity identification and recognition from data captured from various kinds of sensors, including static sensors, wearable equipment, and wrist worn *RFID* readers, etc. Environment sensors such as infrared motion detectors and magnetic door sensors have been used to gather information about complex activities such as cooking, sleeping and eating. These sensors are used in performing location-based activity recognition in indoor environments [21–23]. Ambulatory movements like walking, running, sitting, climbing and falling can be recognized from captured data through wearable sensors [24,25]. Meanwhile, many kinds of approaches have been developed to recognize activities from raw data including Naive Bayes classifiers [26,27], Hidden Markov models [28] and Decision trees [25]. Discriminative approaches including support vector machines and conditional

random fields have also been effective [22]. Unsupervised discovery and other recognition methods have also been introduced [29]. Transfer learning is studied under a variety of different names including learning to learn, life-long learning, knowledge transfer, inductive transfer, context-sensitive learning and meta learning for the purpose of activity recognition [30–32]. An approach is proposed to recognize multi-user activities by using Body Sensor Networks [37]. Different to the traditional data-centric methods for activity recognition, [33] introduces a knowledge-driven approach to real-time, continuous activity recognition based on multisensory data streams in smart home. Behavior patterns in active daily life are more useful than only activities, some methods to discovery activities can be used to identify behavioral patterns in observational sensor data [34,36]; another methods to mine and to monitor human activity patterns have been developed for home-based health monitoring systems [35]. All these works on activity identification and recognition lay a foundation for (i) data analysis and integration, and (ii) information extracting and explaining, in the Internet of Things. However, most of these works only focus on how to identify the isolated activities leveraging the smart contexts information, while relatively few works focus on information extracting (including both the events and their internal semantic links) from massive and heterogeneous data generated from objects in the Internet of Things.

Many semantic Web technologies such as RDF,<sup>1</sup> SPARQL<sup>2</sup> and OWL,<sup>3</sup> have been well developed for querying, browsing and reasoning semantic data. Some models are proposed to build semantic links, such as Linked Data which aims to build semantic links between data on the semantic web according to the internal relationship [38]. SLN schema is an emerging model in semantic network which provides a theory of normalized forms to manage and query on the complicated network [39,40]. Though those semantic tools or works are well developed or proposed, this does not eliminate the important role of information extracting in WoT where huge amounts of data are required to be timely processed. Integrating, fusing and interpreting data as well as extracting meaningful information from raw data are still big stones along the road to achieve the vision of WoT [41].

Some event models have already been proposed to extracting information based on different definitions in different domains. In the Topic Detection and Tracking (TDT) domain, an event model can be defined as a group of themes [42], while in the webpage domain, an event model (e.g. DOM model) is a series of operations related to the browser GUI such as pressing a button, clicking a hyperlink and submitting a form.<sup>4</sup> Some other event models are defined based on several properties such as LODE model [43] (one class and six properties) and EO model [44] (four classes and seventeen properties). Most of these definitions are based on a few characteristics. In our work, we try hard to construct the Web of Events and we propose our model on a series of comprehensive concepts such as event, event type, link type and event schema rather than a few limited characteristics, which are more suitable for the WoT scenarios where myriads of data generated all the time. So we redefine an event and identify it as an abstraction from a series of data which is more meaningful than data. And we demonstrate how an event is extracted as well as how to construct a link web in large sections.

### 3. Conceptions: Event, event type, link type, and event schema

An event is an action which occurs in the physical world and is usually accompanied by a set of correlative status changes. The monitoring devices can capture the data about the status changes when

---

<sup>1</sup><http://www.w3.org/RDF/>.

<sup>2</sup><http://www.w3.org/TR/sparql11-query/>.

<sup>3</sup><http://www.w3.org/Submission/2006/10/>.

<sup>4</sup><http://www.w3.org/DOM/>.



Fig. 1. An example event of *Having class*.

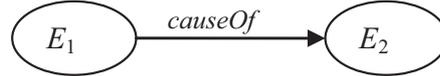


Fig. 2. An example of semantic link between events.

an event occurs in the physical world, and then transmit these data to the cyber world. The captured data should be encapsulated into units according to events specification for facilitating and promoting the understanding to the data through information extracting. In the cyber space, an event can be represented as a list of data.

**Definition 1.** An event is a tuple with the format like  $E(\text{timeDescription}, \text{ObjectsList}, \text{LocationDescription}, \text{DataList})$ , where *timeDescription* records the time when the event occurs, the starting and ending time or duration, *ObjectsList* ( $Oid_1, Oid_2, \dots, Oid_n$ ) is a list of objects involved in the event, *locationDescription* records the location where the event occurs, and *DataList* is a list of data describing the event-related status or interrelations of the objects in *ObjectsList*. In the Internet/Web of Things, the *DataList* might be a summary of the raw data captured across time and space.

For example, we can use  $e20$  (*Having class, 2012-03-2620:41:31, Classroom 8, Wen Liang, 0d0h53m*) to depict an event “Wen Liang has a class at 20:41:31 2012-03-26 for 53 minutes at the place of Classroom 8” where the data in the list is filtered from the raw data. Event information like this is more intuitive, easy-to-understand and easy-to-use than raw data for users or intelligent systems.

Often, not all data sensed from the physical world is useful to restore the event information. Moreover, not all events are useful for users to grasp the situation or to solve problems. An intuitional idea is to extract useful event information from the detailed raw data based on well-defined event types provided by the domain experts or intelligent mining tools.

There are various types of events in our daily life, and for events with the same type, similar solutions can be adopted due to their common characteristics. In the cyber space, an event type can be defined as a model to formulate the data of the event instances with the same type.

**Definition 2.** An event type  $ET$  is a 2-tuple (*Precondition*, *EventDescription*), where *Precondition* is a logic expression  $(P_{11} \wedge P_{12} \wedge \dots \wedge P_{1k_1}) \vee (P_{21} \wedge P_{22} \wedge \dots \wedge P_{2k_2}) \vee \dots \vee (P_{n1} \wedge P_{n2} \wedge \dots \wedge P_{nk_n})$ , each part of which is a condition to the event. *EventDescription* is a data record consisted of name and descriptions of the event like (*timestamp*, *ObjectsList*, *LocationDescription*, *DataList*) as defined in Definition 1.

The event types are used to regularize the events which can be mined from the massive and heterogeneous data. When the captured data meets the condition *Precondition* in Definition 2, an event of the defined  $ET$  type is confirmed, and the event information *EventDescription* would be posted on the web.

For example, the event type of *Having class* can be defined as follows:

The Event of *Havingclass*:  $((\text{duration} \geq 45 \text{ mins}) \wedge (\text{locationName} = \text{Classroom}), (\text{start\_time}, \text{end\_}$

*time, duration, location, Having class*))

While an mobile device reports a measurement (latitude = 39.9661140441895, longitude = 116.372123718262) of a person (with a *objectID* = “867495012228536”) to the system at 2013-03-26 20:41:31, the agent can search some necessary information of the people from the Resource Repository according to *objectID* like *UserID* = 200911211015, *Location* = Classroom 8 and *name* = Wen Liang, then determines whether the precondition is satisfied for the event type of *Having class*. The agent would name the event and post it on the web in the form shown in Fig. 1.

Teachers or other users can access the event information through the web and understand the event: “Wen Liang has a class at 20:41:31 2012-03-26 for 53 minutes at the place of Classroom 8”.

**Definition 3.** A link between two events is a 3-tuple  $L(startEvent, endEvent, factor)$ , which means that there is a *factor* relation linking *startEvent* to *endEvent*.

For example, *time sequence* (like *direct-succeeding*) and *causeOf* are two general kinds of links between smart campus events.  $(E1, E2, causeOf)$  means that an event *E1* *causes* another event *E2*, which can be shown in Fig. 2.

Links between events weave the isolated events into a connected network which might be more useful than a set of isolated events and provides a global view of the situation for users by letting them grasp the internal relationships between events more easily. More useful evolving patterns can be mined from the network which can be used to predict future trends in the field, and potential relations between events can be deduced based on the existing links. For example, we can find the inherent reasons of some illness event by the transitivity of the *causeOf* links among a series of events.

Indeed, there are various kinds of link types with different factors. The link types can be determined by a pair of event types, that is, between a pair of event types there are some certain link types with certain factors.

**Definition 4.** A link type *LT* between event types can be represented as  $LT(startEventType, endEventType)$ , which means there is a link type from event type *startEventType* to event type *endEventType*. The link factor between two events can be defined by some certain inherent properties of the events.

A set of link types can be defined according to the requirements of a certain domain. For two certain events, the semantic links between them can be determined by their inherent properties. For example, a *direct-succeeding* link between an event of *Having class* and another of *Having lunch* is determined by the end time of the lunch event and the start time of the class event. The filters to determine the semantic links between events can be worked out according to the actual applications. Even more, there might be a series of reasoning rules among these link types which can be used to deduce the potential internal relationships implied in the existing links. For example, the link type *causeOf* has the character of transitivity, i.e.,  $causeOf * causeOf \rightarrow causeOf$ .

For a given application, experts can define a set of useful event types, and a set of link types as well as a set of reasoning rules based on the link types. These well-defined event types, link types and the reasoning rules construct a domain- or application-dependent event schema. Indeed, an event schema is a domain-dependent knowledge base to differentiate the critical and sensitive information from the massive and heterogeneous data.

**Definition 5.** An event schema has 3 components: (i) a set of event types  $\{ET_1, ET_2, \dots, ET_n\}$  where each  $ET_i$  is an event type, (ii) a set of link types  $\{LT_1, LT_2, \dots, LT_m\}$  where each  $LT_k$  is defined between a pair of  $(ET_i, ET_j)$ , and (iii) a set of reasoning rules  $(r_1, r_2, \dots, r_l)$  where each  $r$  is in the form of  $LT_i * LT_j \rightarrow LT_k$ .

Event schemas may vary depending on different application scenarios and play the most important role during the extracting process. Traditionally, the event schema should be defined by the domain experts or mined from a large scale of history information.

## 4. A model for event information extracting

### 4.1. Framework of the Web of Events

Figure 3 shows the framework of the Web of Events. In the Internet of Things, trillions of smart things are distributed in different types of application scenarios such as e-Health, logistics and smart homes. The event extracting agents in these application scenarios extract various meaningful events constantly from the captured large scale raw data according to the local event schemas provided by domain experts. The derived events would be transmitted to the Web of Events. The Web of Events can organize the events in a multi-classification way and be capable of integrating the events from different scenarios seamlessly for specific purposes. The Web users or intelligent agents can access to the events easily through REST related technologies like HTTP or other mobile protocols.

In this framework, there are two key challenges: (i) How to extract the event information from the raw data. The event schema consisting of a series of event types plays the most important role during the extracting process. Indeed, the event schema is a base of knowledge derived from the domain experts or intelligent knowledge mining tools. (ii) How to organize and integrate events from different application scenarios. These events would be endowed with enriched semantics. Besides, the various semantic relationships among the events would be helpful to weave an event network with semantic links which would be more useful for implementing the intellectualization of the Internet of Things. This paper focuses on the former challenge: *the event extracting*.

### 4.2. Model for event information extracting

The process of extracting event information consists of three phases: (i) collecting raw data by sensing devices, (ii) parsing the captured data and extracting event information, and (iii) organizing and presenting event information on the web. Therefore, the model for event information extraction is 3-layered architecture as shown in Fig. 4. The data collecting layer aims at capturing raw data from different kinds of application scenarios with the support of smart sensors and transmitting the data to the upper layer with some specific formulation; the event extracting layer encapsulates raw data with the basic information of related objects stored in the system and extracting the event information with the help of the event schema; and the aims of the presenting layer lie in two aspects: 1) organizing and integrating the events in a way of category and semantics; 2) providing a platform for Web users and intelligent agents to access the events.

While the data flow continuously pipes into the system, the most important point is to extract useful information by encapsulating the interrelated data in the event extracting layer. The details about the model are discussed as follows.

#### 4.2.1. Data collecting layer

In the data collecting layer, sensors, RFID readers or other monitoring devices collect data from the physical world in time, and transmit the collected data to *Data Agent* in the event extracting layer with

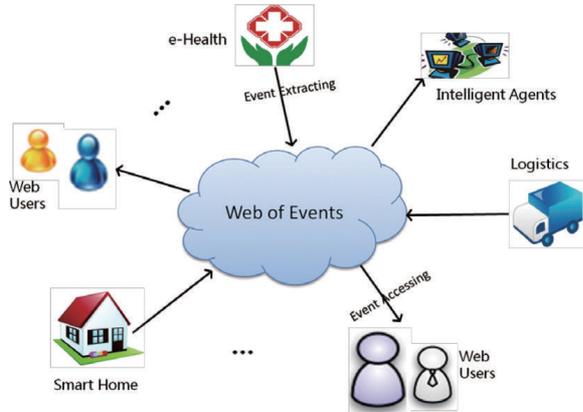


Fig. 3. A framework of the Web of Events.

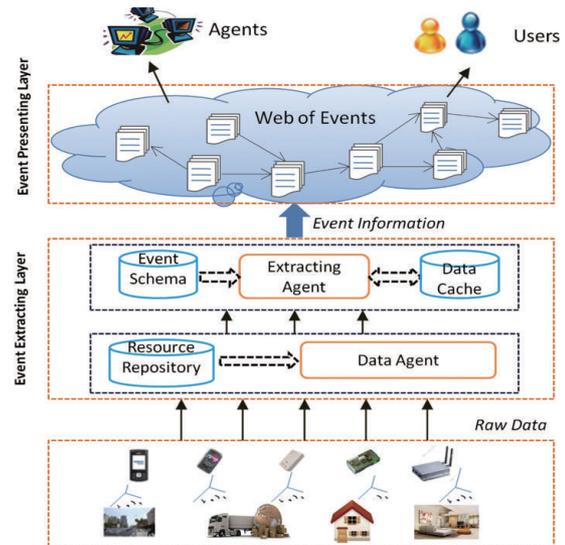


Fig. 4. The model for event information extracting from raw data in WoT.

specific formats upon device types. The timestamp, target object ID and captured content (such as *temperature*, *humidity*, *location* or other measurements) are required in each item of the data. Each monitoring device is endowed specified functions by the manufacturers and the engineers of the applications.

The collected data formats depend on the monitoring devices in specific applications. In most cases, the data is massive and heterogeneous due to the various kinds of different devices and the long-term continuously detection. Critical and sensitive information about useful events is drowned in these trivial and complex data.

#### 4.2.2. Event extracting layer

The event extracting layer aims at extracting useful event information from the massive and heterogeneous data, with the support of five components: *Resource Repository*, *Data Agent*, *Data Cache*, *Extracting Agent* and *Event Schema*.

1. *Resource Repository* stores the primary inherent information like name, manufacturer, timestamp and other perpetual attributes of each monitored objects in the application. The perpetual information might support the event extracting process. In practice, the resource repository might be put on the web or in the cloud store service.
2. *Data Agent* is responsible for data preprocessing including analyzing the source, target object, explaining semantic content of the raw data, searching out the basic information of the object from the repository and shunting the well-processed data to the extracting agent.

In the Internet of Things, each target object in the physical world would be identified by a unique ID. Usually, each record in raw data from the data collecting layer can be transformed into the format like  $\{Oid, DataType, Value, Timestamp\}$ , where *Oid* represents the target object, *DataType* is the type of the value which specifies the expected purpose of the monitored data, *Value* is the monitored value of the target object, and *Timestamp* records the time when the data is generated. In line with the unique *Oid*, each record of the data can be merged with the corresponding inherent information of the object stored in *Resource Repository*, and then the data record could be encapsulated into a specific format like (*Oid*,

*DataType*, *Value*, *Timestamp*, *InformationList*), where *InformationList* is a list of inherent information of the target which is useful to the generated events. Then the encapsulated data would be transmitted to the extracting agent for further processing.

The event schema stores information of all possible event types and link types in the application each of which includes a precondition.

1. *Data Cache* stores history data from data agent which might be used in the following processing. The operating principles for the extracting agent are listed as follows.
2. Once some data records arrive, the *Extracting Agent* first make sure the relevant event types which the data might trigger. There might be not only one event type related to the coming data. For each possible event type, the agent judges whether the precondition is satisfied, sometimes with the support of the data cache which can provide the necessary history data. During this judgment, the agent produces a new event with the provided data if the data meets the precondition and transmits the event information to the event presenting layer. Otherwise, the data would be put into *Data Cache* if they might be useful in the following analysis. Or the data would be thrown away.
3. *Event Schema*. Obviously, only the critical and sensitive data meeting the conditions of the event types would be utilized to generate events. Most data would be thrown away after the extraction. The semantics of the generated events is more easy-to-understand than the raw data. The process of the event extracting can somehow be viewed as a process of information discovery and data explanations.

Once the events have been extracted, we can check the internal links between each pair of events according to the predefined link types and their filters. Then, a network of event is generated with a set of semantic links among events.

#### 4.2.3. Event presenting layer

The event presenting layer organizes the events according to their types, timestamp and the internal links among them or other information, and presents the event information in intuitive ways to users through web tools like HTTP or to intelligent systems with general interfaces for the purpose of decision making or problem solving automatically and autonomously.

## 5. Case study – web of smart campus events

The advent of the Internet/Web of Things enables a lot of new application scenarios in business and daily life. This section introduces a case study to show the applicability of the proposed model presented above.

### 5.1. Scenario

A smart campus is an intelligent agent that perceives the state of resident and the physical environments using various kinds of devices like mobile phones, sensors and readers. These devices can capture massive concrete data about individual's activities, environment settings and inhabitants' characteristics. In this section, we'd like to illustrate how to construct the Web of Events through the proposed models.

On the one hand, to better verify our model's performance in dealing with heterogeneous data, we build a student attendance system in our laboratory. One reader as well as eight antennas is installed and these antennas are deployed in eight regions of two rooms. 14 RFID tags are assigned to 14 students respectively. On the other hand, we develop an android program to collect two person's daily events manually and their GPS every five minutes automatically. Therefore, we collect two diverse datasets from these two different scenarios for about one week.

Table 1  
Event types in the mobile scenario

Event type	Conditions and stored information	Descriptions
<i>Having breakfast</i>	$((6:30 < \text{startTime} < 9:30) \wedge (\text{duration} \geq 5 \text{ mins}) \wedge (\text{locationName} = \text{Canteen}), (\text{start\_time}, \text{end\_time}, \text{duration}, \text{location}, \text{Having breakfast}))$	<i>Having breakfast</i> is the type of events which occurs in the canteen (obtained from GPS, similarly hereinafter) lasting for no less than 5 mins and the start time is between 6:30 and 9:00 (based on school timetable).
<i>Having lunch</i>	$((10:30 < \text{startTime} < 13:00) \wedge (\text{duration} \geq 20 \text{ mins}) \wedge (\text{locationName} = \text{Canteen}), (\text{start\_time}, \text{end\_time}, \text{duration}, \text{location}, \text{Having lunch}))$	<i>Having lunch</i> is the type of events which occur in the canteen lasting for no less than 20 mins and the start time is between 10:30 and 13:00 (based on school timetable).
<i>Having supper</i>	$((16:30 < \text{startTime} < 18:30) \wedge (\text{duration} \geq 20 \text{ mins}) \wedge (\text{locationName} = \text{Canteen}), (\text{start\_time}, \text{end\_time}, \text{duration}, \text{location}, \text{Having supper}))$	<i>Having supper</i> is the type of events which occur in the canteen lasting for no less than 20 mins and the start time is between 16:30 and 18:30 (based on school timetable).
<i>Playing sports</i>	$((\text{duration} \geq 30 \text{ mins}) \wedge (\text{locationName} = \text{Playground}), (\text{start\_time}, \text{end\_time}, \text{duration}, \text{location}, \text{Playing sports}))$	<i>Playing sports</i> is the type of events which occur in the playground lasting for no less than 30 mins.
<i>Having class</i>	$((\text{duration} \geq 45 \text{ mins}) \wedge (\text{locationName} = \text{Classroom}), (\text{start\_time}, \text{end\_time}, \text{duration}, \text{location}, \text{Having class}))$	<i>Having class</i> is the type of events which occurred in the classroom lasting for no less than 45 mins (one class).
<i>Studying</i>	$((\text{duration} \geq 30 \text{ mins}) \wedge (\text{locationName} = \text{Library}), (\text{start\_time}, \text{end\_time}, \text{duration}, \text{location}, \text{Self-Studying}))$	<i>Self-Studying</i> is the type of events which occur in the library lasting for no less than 30 mins.
<i>Working</i>	$((\text{duration} \geq 30 \text{ mins}) \wedge (\text{locationName} = \text{Laboratory}), (\text{start\_time}, \text{end\_time}, \text{duration}, \text{location}, \text{Working}))$	<i>Working</i> is the type of events which occur in the laboratory lasting for no less than 30 mins.
<i>Being in the dorm</i>	$((\text{duration} \geq 30 \text{ mins}) \wedge (\text{locationName} = \text{Dormitory}), (\text{start\_time}, \text{end\_time}, \text{duration}, \text{location}, \text{Being in the dorm}))$	<i>Working/Relaxing</i> is the type of events which occur in the dormitory lasting for no less than 30 mins.

## 5.2. Event schema for smart campus

Our schema depends on the real dataset. Firstly, we work out the Event Schema of Smart Campus which involves in a series of event types: *Having breakfast*, *Having lunch*, *Having Supper*, *Entertainment*, *Discussion*, *Playing sports*, *Having class*, *Studying*, *Working*, *Being in the dorm*, *Leave temporarily*, *Being absent*, *Leaving early* and *Being late*. The last four event types are suitable for detecting students' attendance events, while the others are for daily events in campus.

Time range for each event type is necessary to identify an event. But in different scenarios, time ranges for each event type could be different. Thus, we define time range based on school timetable or laboratory timetable. The event types in the mobile scenario and RFID scenario are listed in Tables 1 and 2 respectively.

All event types are defined in advance, and the conditions of event types except *Entertainment* and *Discussion* are shown in Table 1. *Entertainment* and *Discussion* used in the android program are defined by us too, but a concrete event is selected by user according to specific circumstance. As for automatically collected data, we gain event types according to users' submitted information and Table 1 (if submitted information is missing).

Then we can work out possible link types among the above event types. Herein, for the consideration of simplification, only five link types (*d-succeeding*, *co-occur*, *overlap*, *sameTypeOf* and *causeOf*) between any two event types are defined according to the actual data as shown in Table 3, where we also

Table 2  
Event types in the RFID scenario

Event type	Conditions and stored information	Descriptions
<i>Leave temporarily</i>	$((8:30 < start\_Time, end\_Time < 11:30) \vee (14:00 < start\_Time, end\_Time < 16:40)) \wedge (duration \geq 10 \text{ mins}) \wedge (locationName = \text{Out of Laboratory}), (start\_time, end\_time, duration, location, Leave temporarily))$	<i>Leave temporarily</i> is the type of events which occur out of the laboratory (monitored by a RFID reader, similarly hereinafter) lasting for no less than 10 mins, and both the start time and the end time are within the range of 8:30 ~ 11:30 in the morning or 14:00 ~ 16:40 in the afternoon (based on laboratory timetable).
<i>Being absent</i>	$((start\_Time < 8:30) \wedge (end\_Time > 11:30) \vee (start\_Time < 14:00) \wedge (end\_Time > 16:30)) \wedge (locationName = \text{Out of Laboratory}), (start\_time, end\_time, duration, location, Being absent))$	<i>Being absent</i> is the type of events which occur out of the laboratory and both the start time and the end time are not in the range of 8:30 ~ 11:30 and 14:00 ~ 16:40 (based on laboratory timetable).
<i>Leaving early</i>	$((start\_Time < 11:30 < end\_Time) \vee (start\_Time < 16:40 < end\_Time)) \wedge (duration \geq 30 \text{ mins}) \wedge (locationName = \text{Out of Laboratory}), (start\_time, end\_time, duration, location, Leave temporarily))$	<i>Leave temporarily</i> is the type of events which occur out of the laboratory (monitored by a RFID reader) lasting for no less than 30 mins and the start time is earlier than 11:30 in the morning or 16:40 in the afternoon (based on laboratory timetable).
<i>Being late</i>	$((start\_Time < 8:30 < end\_Time) \vee (start\_Time < 14:00 < end\_Time)) \wedge (locationName = \text{Out of Laboratory}), (start\_time, end\_time, duration, location, Being late))$	<i>Being late</i> is the type of events which occur out of the laboratory (monitored by a RFID reader) and the end time is later than 8:30 in the morning or 14:00 in the afternoon (based on laboratory timetable).

list the descriptions of specific link type and its corresponding format. Meanwhile, we pose the judgment condition of specific link type, so that by using the conditions links can be generated by system automatically.

### 5.3. Architecture of the prototype

To verify the event extraction model, we have built a prototype using Tomcat as the Web server and MySQL5.5 as the database. Figure 5 shows the overall event extraction process in a simplified Smart Campus application.

In the data collection layer, we gain the captured RFID data flow from the attendance system and mobile data from android procedure ①. The attendance system captures RFID binary data all the time and transforms data into the format of  $\{Oid, DataType, Activeid, Timestamp\}$ , where *Oid* is RFID tag to identify the target person, *DataType* is used to describe that the captured data is from RFID source, *Activeid* is detected antenna ID when a person enters corresponding region, and *Timestamp* indicates the measured time. The preprocessed RFID data will be sent to the extracting layer in JSON format, which maybe more adapted to devices with limited capabilities [45].

The format of mobile data recorded by person manually is  $\{Oid, DataType, Etype, Title, Note, Latitude, Longitude, Timestamp\}$ , where *Oid* is the serial number of the target mobile, *DataType* is used to describe that the captured data is from mobile source, *Etype* is the general event type defined by users, *Title* and *Note* is event title and content of an event recorded by a user, *Latitude* and *Longitude* is GPS information recorded automatically, and *Timestamp* indicates the recorded time. Besides, a person's GPS information will be captured every five minutes automatically to excavate those unrecorded events man-

Table 3  
Link types between event types in smart campus

Link type	Format	Conditions	Descriptions
<i>D-succeeding</i>	$(e_1, e_2, d\text{-succeeding})$	$((e_1.end\_time < e_2.start\_time) \wedge (e_2.start\_time < e_1.start\_time))$ , where $e$ is any event such that $e.start\_time > e_1.end\_time$ .	$e_2$ is a directly succeeded event after $e_1$ ends.
<i>Co-occur</i>	$(e_1, e_2, co\text{-occur})$	$((e_1.start\_time < e_2.start\_time) \wedge (e_1.end\_time > e_2.end\_time) \vee ((e_2.start\_time < e_1.start\_time) \wedge e_2.end\_time > e_1.end\_time))$	$e_1$ occurs in the period of $e_1$ , vice versa.
<i>Overlap</i>	$(e_1, e_2, overlap)$	$((e_1.start\_time < e_2.start\_time) \wedge (e_1.end\_time > e_2.start\_time) \wedge (e_1.end\_time < e_2.end\_time) \wedge (e_2.start\_time < e_1.start\_time))$ , where $e$ is any event such that $((e.start\_time > e_1.start\_time) \wedge (e.start\_time < e_1.end\_time) \wedge (e.end\_time > e_1.end\_time))$	There is an overlap period between the periods of $e_1$ and $e_2$ .
<i>SameTypeOf</i>	$(e_1, e_2, sameTypeOf)$	$(e_1.type = e_2.type)$	$e_1$ and $e_2$ are with the same event type.
<i>CauseOf</i>	$(e_1, e_2, causeOf)$	$e_1.start\_time < e_2.start\_time$ , and other conditions should be defined according to the event types of $e_1$ and $e_2$ .	$e_2$ is caused by $e_1$ . For example, A person is absent because he goes to play sports.

ually. The format of GPS information is  $\{Oid, DataType, Latitude, Longitude, Timestamp\}$ . Both kinds of mobile data will be sent in JSON format.

In the event extracting layer, the data agent process ② will first search out the basic information of each data record according to relative unique *Oid* from a person information Table ③. Information of latitude and longitude will be converted into specific location. Then each data record will be encapsulated into the format like  $\{Oid, DataType, Value, Timestamp, Location \dots\}$ , and the well processed data list will be shunted to the extracting agent process ④.

The extracting agent process undertakes the most important work of event extracting. Firstly, it will find out the relative event types according to the data type of each data record from a mapping Table ⑤. Then, for each event type of each data record, it will call two main sub-process modules – the data caching module ⑥ and the event generating module ⑦. Each event type has a data caching table. The data caching is responsible for data caching and labeling the changing state of data as well as its starting and ending time from which we can calculate the duration time of an event. The event generating module decides whether and what event should be generated according to the changing state of data. Both modules are sharing a group of event types ⑧.

In the event presenting layer, the generated events in the event table will be presented on web pages with the help of corresponding processes ⑨. Figure 8 shows the final pages to users, where the left web page shows the event list. These events are archived according to dates and can be accessed by event name, event type, timestamp, objected or name. The detailed event information of one specific event is showed on the right web page.

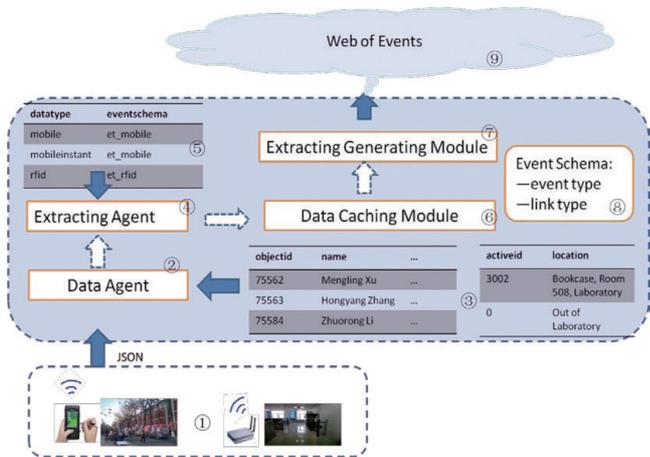


Fig. 5. Architecture of the prototype.

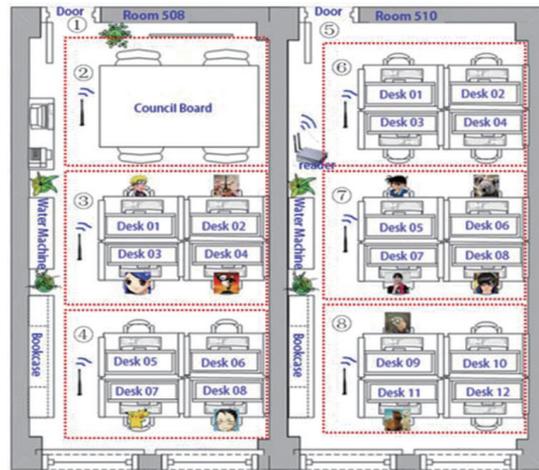


Fig. 6. The student attendance system.

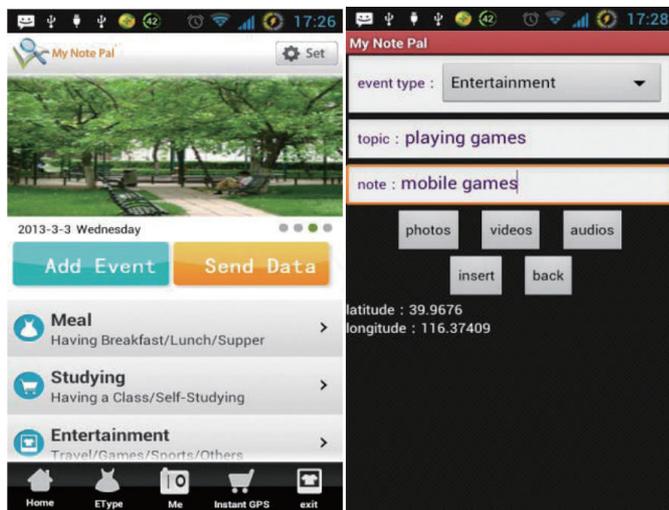


Fig. 7. Using android procedure for collecting data.

Table 4

Event counts of nine event types

EventType	Counts
Being absent	3
Being late	1
Being in the dorm	11
Discussion	3
Entertainment	27
Having class	18
Having breakfast	9
Having lunch	11
Having supper	14
Leaving early	1
Playing sports	1
Studying	23
Working	5
Total	127

### 5.4. Implementation and presentation

#### 5.4.1. Data collecting

We build a student attendance system in our laboratory (See Fig. 6). The system has a reader installed in Room 510, eight antennas installed in eight regions (1–8) of Room 508 and 510, and 14 active RFID tags assigned to 14 persons. When a person with a RFID tag is entering region 8, the reader will detect the activated RFID tag and send data (incl. tag ID and antenna ID) to the host computer in the binary format. The reader can detect the person’s location according to the corresponding antenna (active id: 3006). Then the binary data will be processed and sent to the extracting server in a JSON format, like: [{"oid": "75592", "datatype": "rfid", "activeid": "3008", "timestamp": "2013-03-07 16:54:30"}].

We also develop an android program for collecting persons’ daily events (See Fig. 7). One can record

The screenshot displays the 'Web of Smart Campus Events' interface. The main page features a navigation bar with 'homepage', 'events network', 'schema', and 'about'. Below this is a 'Sort By' section and a 'Recent Event Information' table. The table lists events with columns for EventNumber, Location, EventType, Name, and Timestamp. A red arrow points from the 'Detail' link for event e20 to a detailed view on the right. The detailed view shows: 'Wen Liang is having class, at Classroom 8.', 'EventNumber: e20', 'EventType: having class', 'Timestamp: 2013-03-26 20:41:31', 'UserID : 200911211015', 'Name: Wen Liang', 'Location: Classroom 8', and 'Duration: 0d,0h,53m'.

EventNumber	Location	EventType	Name	Timestamp
e12	Laboratory	Entertainment	Zhuorong Li	2013-03-25 22:15:16
e19	Canteen JiaoGong	Studying	Wen Liang	2013-03-26 14:20:11
e20	Classroom 8	having class	Wen Liang	2013-03-26 20:41:31
e22	Laboratory	Having supper	Wen Liang	2013-03-26 22:44:07
e23	Laboratory	Entertainment	Wen Liang	2013-03-26 22:49:29
e24	Canteen Xinlequn	Entertainment	Wen Liang	2013-03-26 23:17:52
e28	Laboratory	Having breakfast	Wen Liang	2013-03-27 08:35:04

Fig. 8. Interface for smart campus event presentation.

his daily events manually, while his GPS information will be captured automatically every five minutes. These two kinds of mobile data will be sent in JSON format automatically, like:

```
[{"oid":"352274016686995","datatype":"mobile_noteinstant","timestamp":"2013-04-02 17:16:33",
"longitude":116.37403869628906,"latitude": 39.96767807006836}]
[{"oid":"352274016686995","datatype":"mobile_note","etype":"Entertainment","topic":" playing
games", "note":"mobile games", "timestamp":"2013-04-02 17:28:26", "longitude": 116.3740925781
25, "latitude":39.96764155883789}]
```

To make the data collecting layer more flexible and more suitable, we turn the physical devices (RFID tag or mobile) into RESTful resources and offer a uniformed REST API to access these devices. Request the resources will be formulated using a standard URL. For example, we can contact the root URL with the verb POST to add a mobile node: <http://ireg.bnu.edu.cn/WoEServer/datacollect/mobile/addnode/>.

We can delete the mobile node whose oid is 352274016686995 by contacting the URL:  
[http:// ireg.bnu.edu.cn/WoEServer/datacollect/ mobile /deletenode/352274016686995](http://ireg.bnu.edu.cn/WoEServer/datacollect/mobile/deletenode/352274016686995)

The following URL are used to communicate mobile data and GPS information with the server respectively:

[http:// ireg.bnu.edu.cn/WoEServer/datacollect/ mobile /data](http://ireg.bnu.edu.cn/WoEServer/datacollect/mobile/data)

[http:// ireg.bnu.edu.cn/WoEServer/datacollect/ mobile /instantdata](http://ireg.bnu.edu.cn/WoEServer/datacollect/mobile/instantdata).

Table 5  
Link counts of five link types

Link Type	Counts
d-succeeding	100
co-occur	43
overlap	3
sameTypeOf	114
causeOf	0
Total	260

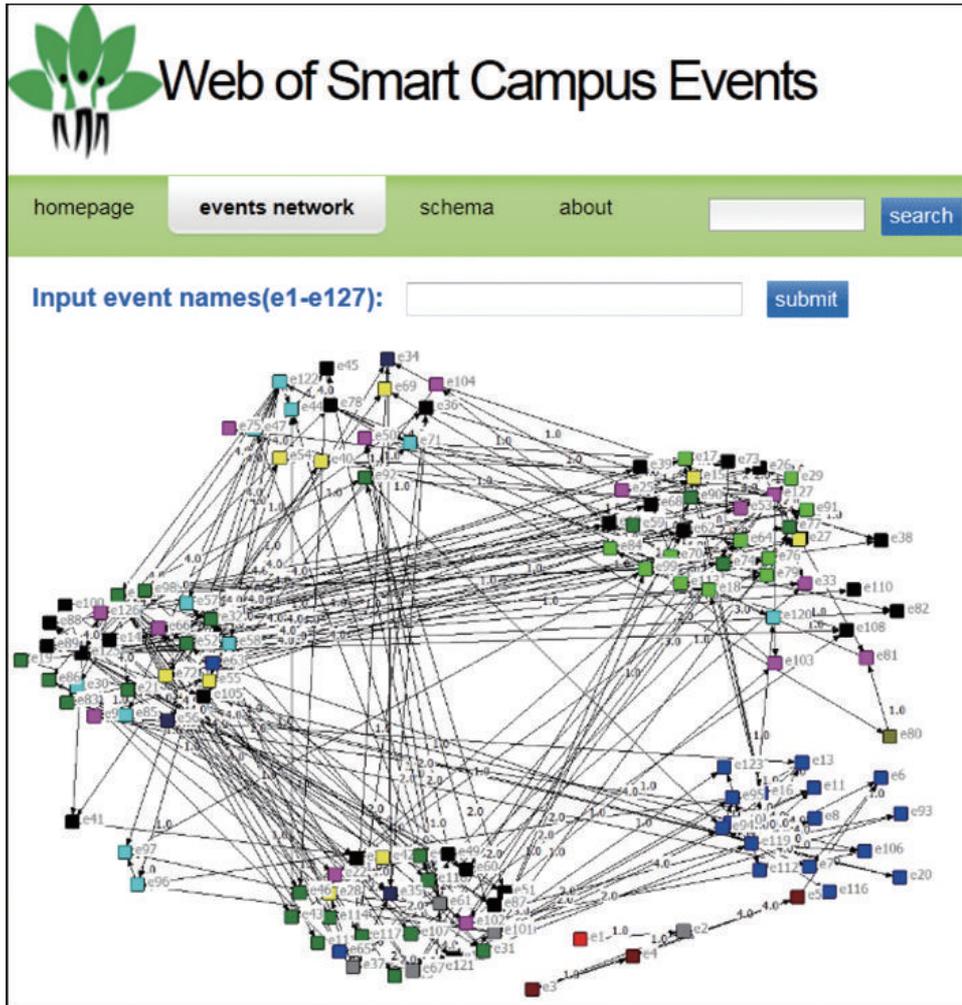


Fig. 9. Interface for smart campus event link presentation.

#### 5.4.2. Event extracting and presentation

At the beginning of the extracting layer, Data Agent will find out specific location information of the submitted data. For RFID data, we can gain location according to the active id of each data record from a mapping table. For mobile data, the indoor error of latitude and longitude is greater than outdoor error as GPS signal is unavailable in the room. Through experiment we find that the error is stable in

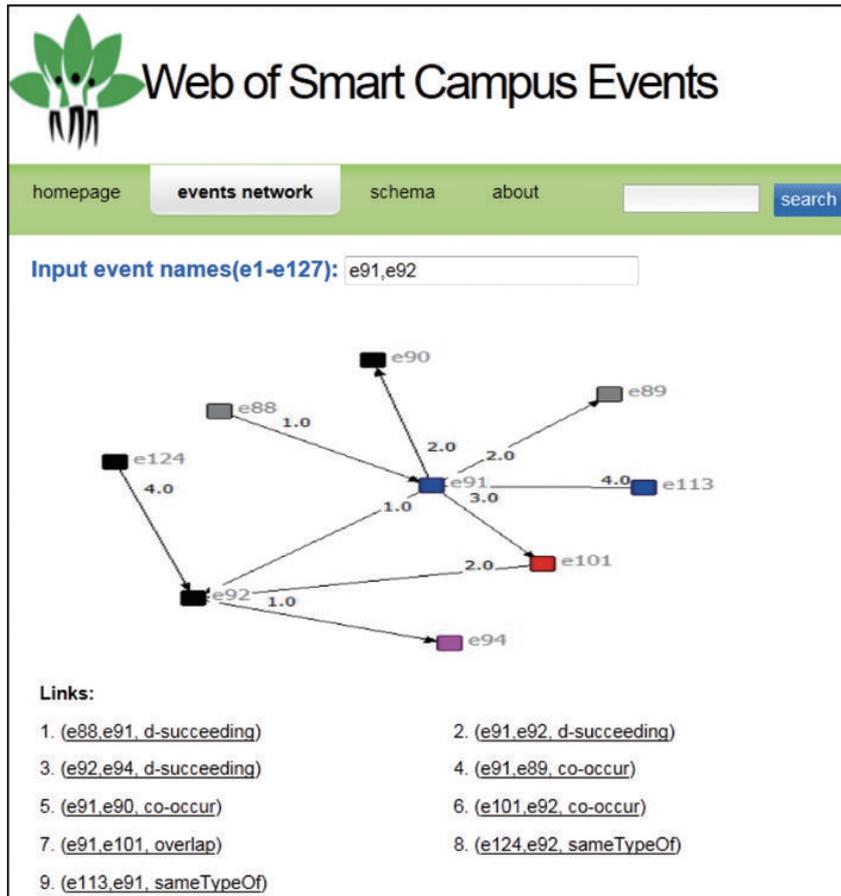


Fig. 10. Query results: links connected to two selected events e91 and e92.

the same location. So we randomly select a group of points (with latitude and longitude) to represent the corresponding location. By calculating whether a target point belongs to the represented points, we can find the target location. Then the data will be converted into the format of (*oid*, *datatype*, *location*, *timestamp*, *duration*, *name*).

```
{"oid": "352274016686995", "datatype": "mobile_note", "location": "Laboratory", "timestamp": "2013-04-02 17:28:26", "name": "Zhuorong Li"}
```

We have developed a procedure to extract the event information according to the above defined event types. Total 127 events of 13 types have been extracted from the dataset as shown in Table 4, and 260 links between events of 5 different link types have been retrieved according to the predefined schemas as shown in Table 5. We also developed a tool by using JSP language to present the event information on the Web as shown in Fig. 8.

#### 5.4.3. Link extracting and presentation

We have developed a java procedure to extract the five kinds of link information according to the above defined link types. To well represent link information of the extracted events, we use a social

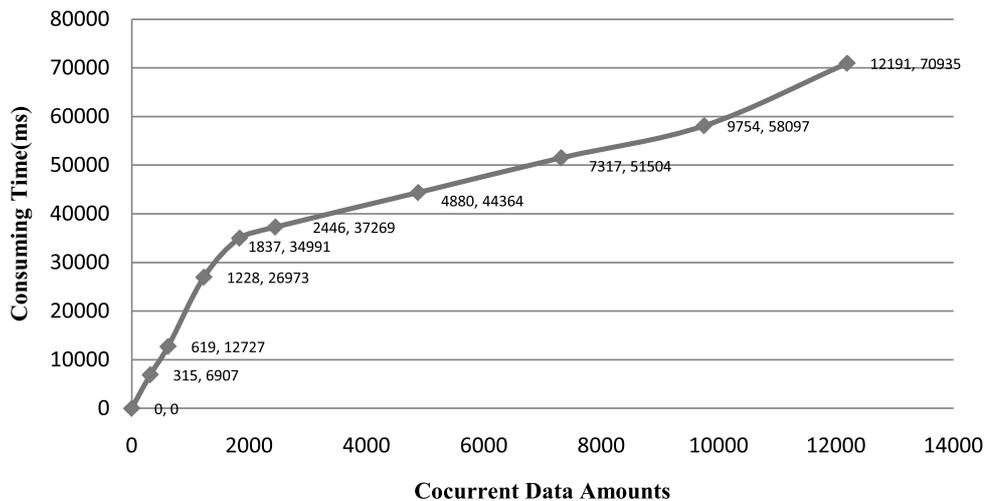


Fig. 11. Consuming time of concurrent data for event extracting layer.

network analysis tool NetDraw<sup>5</sup> for drawing graphs of link information, as shown in Fig. 9, where each node represents an event and each edge represents a semantic link with a link type between nodes. The number on a link reflects corresponding link type.<sup>6</sup> Different colors of nodes reflect different event types. There are almost five groups of nodes in Fig. 9, which reflect the most frequent events happen in five locations. For example, the right groups in color blue reflect having classes in the classroom. While some events are selected, a set of links connected to the selected events can be presented in a way of visualized network or text format as shown in Fig. 10. The developed tool can also provide interfaces to smart devices to support the intelligent decision making systems. Our system offers a uniform RESTful interface. Request the resources will be simply formulated using a standard URL. For example, we can visit some events related to someone with userID 352274016686995 by contacting a URL with the verb GET: <http://ireg.bnu.edu.cn/eventpresent/352274016686995>.

#### 5.4.4. Evaluation

The processing time mainly includes three parts: data transmission time from the physical world to the event extracting layer, data processing time and event extracting time in the event extracting layer, event presenting and link presenting time in the event presenting layer. Data transmission time depends mainly on many external factors such as wireless network and network traffic. Event presenting is not in real-time. So as for a system dealing with real-time data, the key important time performance we should consider is time spending on the event extracting layer. We build our event extracting program on a desktop computer equipped with a 4 G memory, Intel(R) Core(TM) i5-2400 CPU and Windows 7 operating system. Time taken to deal with a mobile data or a RFID data at some time is less than 50 ms.

But in an instant physical environment, myriads of data may be transmitted to the extracting layer at the same time. To simulate this situation, we retain the collected data<sup>7</sup> for one week as much as 12191, including 100 notes recorded by two persons, 689 GPS information recorded by mobile system

<sup>5</sup><http://www.analytictech.com/Netdraw/netdraw.htm>.

<sup>6</sup>1.00 stands for *d-succeeding*; 2.00 stands for *co-occur*; 3.00 stands for *overlap*; 4.00 stands for *sameTypeOf*; 5.00 stands for *causeOf*.

<sup>7</sup>Notice that in the real scenario, raw data will not be retained except the data cache model.

automatically, and 11402 RFID data. We store them separately into three tables and select 9 groups of points as shown in Fig. 11, where we conclude that as concurrent data increases consuming time increases. Our model works well even when the concurrent data is large. For example, consuming time is about one minute when concurrent data amount is 9754. Concurrent data increases almost 1000 (from 2446 to 12191) while consuming time increases less than one minutes.

## 6. Conclusion and discussion

As tremendous data emerging into our life along with the development of the Web of Things, people are more willing to master the situations based on event information than raw data. How to extract the event information and internal links between events by integrating heterogeneous data in the Internet of Things is a key challenge. This paper first introduces the conceptions of *event*, *event type*, *link types* and *event schema*. Then, we propose a framework of the Web of Events to present the events on the Web of Things for the convenience of access to users and intelligent systems. A 3-layered model is proposed for extracting events and their internal relations from large-scale data set based on predefined event schema in a certain domain. The proposed model is an easy-to-understand and easy-to-use tool for filtering useless or meaningless data, so that people can analyze and solve problems by grasping the key events quickly and intuitively. A case study on smart campus is developed to show how to use this model to extract event information (including events and their internal semantic links) from the physical world in an efficient and effective way. The advantages of our approach are as follows:

- The proposed model provides a uniform framework to facilitate the decision-making of people or smart agents by integrating heterogeneous data into understandable event information. Heterogeneous data from different kinds of sensing devices or readers in different places can be used in the event extracting process. Users access the event information extracted from the raw data in a friendly and understandable way.
- The model provides a way to compress massive data generated with the explosive increasing of sensors by neglecting the redundant data. The massive data in the Internet of Things bring several challenges such as how to store, how to manage, how to query, how to process and how to use. The proposed model throws off the redundant data and only reserves the useful and crucial part.
- The Web of Events provides a platform to organize and represent the events extracted from physical world in a friendly and intuitional way. It also provides an efficient tool for further analysis on the interrelated events, especially for the behavior pattern analysis. There might be various kinds of interrelations among the extracted events like sequence (time), cause-effect and consist-of. We can retrieve more potential and implied relations through reasoning on these rich relations. In addition, valuable knowledge among a number of events and their internal links can be mined from raw data and the knowledge would be useful to patterns discovering.

Location is very important for event generating. In our case study, we randomly select a group of points (with latitude and longitude) to represent the corresponding location. There may be other ways to gain location information such as data mining. But which way is better? This problem needs to be resolved in the future efforts. There may be other ways to extract event information such as data mining and ontology. In future, we will try to find other more appropriate ways to extract events as well as improve the performance of our system. In this work, we construct the Web of Events which is simple but flexible enough. In future, we will focus on how to design a reasonable and efficient semantic model of event-event, event-person or person-person based on the emerging technologies of semantic sensor networks domain and continue to explore how to solve problems or to find useful potential patterns with the support of the Web of Events.

## Acknowledgments

This research is supported by National Natural Science Foundation of China (61171014) and the Fundamental Research Funds for the Central Universities.

## References

- [1] D. Guinard and V. Trifa, Towards the Web of Things: Web mashups for embedded devices, in *Proceeding of WWW (International World Wide Web Conferences)*, Madrid, Spain, 2009.
- [2] R. Jain, Eventweb: Developing a human-centered computing system, *IEEE Computer* **41**(2) (2008), 42–50.
- [3] V.K. Singh and R. Jain, Structural analysis of the emerging event-web, in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 1183–1184.
- [4] G. Papamarkos, A. Poulouvassilis and P.T. Wood, Event-condition-action rule languages for the semantic web, *Workshop on Semantic Web and Databases*, 2003, pp. 309–327.
- [5] S.S. Chawathe, V. Krishnamurthy, S. Ramachandran and S. Sarma, Managing RFID data, in *Proceedings of the 30th VLDB Conference*, 2004, pp. 1189–1195.
- [6] J. Liu and A. Terzis, Sensing data centres for energy efficiency, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **370** (January 2012), 136–157.
- [7] S.R. Jeffery, M.G. Garofalakis and M.J. Franklin, Adaptive cleaning for RFID data streams, in *Proceedings of the 32nd international conference on Very large data bases (VLDB)*, 2006, pp. 163–174.
- [8] S.R. Jeffery, G. Alonso, M.J. Franklin, W. Hong and J. Wisom, A pipelined framework for online cleaning of sensor data streams, in *Proceedings of the 22nd International Conference on Data Engineering (ICDE 06)*, 2006, p. 140.
- [9] M. Balazinska et al., Data management in the worldwide sensor web, *IEEE Pervasive Computing* **6**(2) (April 2007), 30–40.
- [10] K.G. Jeffery, The Internet of Things: the death of a traditional database? *IETE Technical Review* **26**(5) (2009), 313–319.
- [11] J. Cooper and A. James, Challenges for database management in the Internet of Things, *IETE Technical Review* **26**(5) (2009), 320–329.
- [12] Y. Diao, D. Ganesan, G. Mathur and P. Shenoy, Rethinking data management for storage-centric sensor networks, in *Proceedings of Third Biennial Conference on Innovative Data Systems Research (CIDR)*, January 2007, pp. 410–419.
- [13] Z. Yu, B. Mo et al., Achieving optimal data storage position in wireless sensor network, *Computer Communications* **33**(1) (Jan. 2010), 92–102.
- [14] M. Palmer, Seven Principles of Effective RFID Data Management, *Progress Software-Real Time Division*, 2004.
- [15] F. Wang and P. Liu, Temporal management of RFID data, in *Proceedings of the 31st international conference on Very large data bases (VLDB)*, 2005, pp. 1128–1139.
- [16] M. Navarro, D. Bhatnagar and Y. Liang, An integrated network and data management system for heterogeneous WSNs, in *Proceedings of 8th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, 2011, pp. 819–824.
- [17] T. Fan and Y. Chen, A scheme of data management in the Internet of Things, in *Proceedings of the 2nd IEEE International Conference on Network Infrastructure and Digital Content*, 2010, pp. 110–114.
- [18] C. Fan et al., A scalable Internet of Things lean data provision architecture based on ontology, in *Proceedings of IEEE GCC Conference and Exhibition*, 2011, pp. 553–556.
- [19] I. Buchan, J. Winn and C. Bishop, A unified modeling approach to data-intensive healthcare, *The fourth paradigm: data-intensive scientific discovery*, 2009, pp. 91–97.
- [20] M. Adnane, Z. Jiang, S. Choi and H. Jang, Detecting specific health-related events using an integrated sensor system for vital sign monitoring, *Sensors* **9**(9) (2009), 6897–6912.
- [21] B. Logan, J. Healey, M. Philipose, E.M. Tapia and S. Intille, A long-term evaluation of sensing modalities for activity recognition, in *Proceedings of the 9th international conference on Ubiquitous computing*, 2007, pp. 483–500.
- [22] T.L. Kasteren, G. Englebienne and B. Kröse, An activity monitoring system for elderly care using generative and discriminative models, *Personal and Ubiquitous Computing* **14**(6) (September 2010), 489–498.
- [23] M. Philipose et al., Inferring activities from interactions with objects, *IEEE Pervasive Computing* **3**(4) (2004), 50–57.
- [24] N.C. Krishnan and S. Panchanathan, Analysis of low resolution accelerometer data for continuous human activity recognition, in *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2008, pp. 3337–3340.
- [25] U. Maurer, A. Smailagic, D.P. Siewiorek and M. Deisher, Activity recognition and monitoring using multiple sensors on different body positions, in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, 2006, pp. 113–116.

- [26] A. Helal et al., The Gator Tech smart house: A programmable pervasive space, *IEEE Computer* **38**(3) (March 2005), 50–60.
- [27] T. van Kasteren and B. Kröse, Bayesian activity recognition in residence for elders, in *Proceedings of the International Conference on Intelligent Environments*, 2008.
- [28] J. Lester, T. Choudhury, N. Kern, G. Borriello and B. Hannaford, A hybrid discriminative/generative approach for modeling human activities, in *Proceedings of the 19th international joint conference on Artificial intelligence*, 2005, pp. 766–772.
- [29] P. Rashidi, D.J. Cook, L.B. Holder and M. Schmitter-Edgecombe, Discovering activities to recognize and track in a smart environment, *IEEE Transactions on Knowledge and Data Engineering* **23**(4) (April 2011), 527–539.
- [30] D.H. Hu and Q. Yang, Transfer Learning for Activity Recognition via Sensor Mapping, in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, 2011, pp. 1962–1967.
- [31] D.H. Hu, X.X. Zhang, J. Xin, V.W. Zheng and Q. Yang, Abnormal Activity Recognition based on HDP-HMM Models, in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI 2009)*, 2009, pp. 1715–1720.
- [32] J. Yin, D.H. Hu and Q. Yang, Spatio-temporal Event Detection Using Dynamic Conditional Random Fields, in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI 2009)*, 1321–1326.
- [33] L. Chen, C.D. Nugent and H. Wang, A Knowledge-Driven Approach to Activity Recognition in Smart Homes, *IEEE Transactions on Knowledge and Data Engineering* **24**(6) (2012), 961–974.
- [34] D.J. Cook, N.C. Krishnan and P. Rashidi, Activity Discovery and Activity Recognition: A New Partnership, *IEEE Transactions on Systems, Man and Cybernetics, Part B (TSMCB)*, 2012, pp. 1–9.
- [35] P. Rashidi and D.J. Cook, A Method for Mining and Monitoring Human Activity Patterns for Home-based Health Monitoring Systems, *ACM Transactions on Intelligent Systems and Technology (TIST), Special Issue on Intelligent Systems for Health Informatics*, 2012.
- [36] T. Gu, L. Wang, X. Tao and J. Lu, A Pattern Mining Approach to Sensor-based Human Activity Recognition, *IEEE Transactions on Knowledge and Data Engineering* **23**(9) (September 2011), 1359–1372.
- [37] T. Gu, L. Wang, X. Tao and J. Lu, Recognizing Multi-user Activities using wearable sensors in a smart home, *IEEE Transactions on Mobile Computing (TMC)* **10**(11) (March 2011), 1618–1631.
- [38] T. Heath and C. Bizer, Linked Data: Evolving the Web into a Global Data Space, *Synthesis Lectures on the Semantic Web: Theory and Technology* **1**(1) (February 2011), 1–136.
- [39] H. Zhuge and Y. Sun, Schema Theory for Semantic Link Network, *Future Generation Computer Systems* **26**(3) (March 2010), 408–420.
- [40] J. Zhang and Y. Sun, An analogy reasoning model for Semantic Link Network, *JDCTA: International Journal of Digital Content Technology and its Applications* **4**(7) (2010), 128–139.
- [41] 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 (IJSWIS)* **8**(1) (2012), 1–21.
- [42] E. Zhou, N. Zhong and Y. Li, Extracting news blog hot topics based on the W2T Methodology, *World Wide Web*, 2013, pp. 1–28.
- [43] R. Shaw, R. Troncy and L. Hardman, Lode: linking open descriptions of events, *The Semantic Web*, 2009, pp. 153–167.
- [44] Y. Raimond and S. Abdallah, The event ontology, 2007. [Online]. Available: <http://purl.org/NET/c4dm/event.owl>.
- [45] D. Guinard, V. Trifa, T. Pham and O. Liechti, Towards Physical Mashups in the Web of Things, in *Proceedings of the 6th International Conference on Networked Sensing Systems (INSS'09)*, 2009, pp. 1–4.

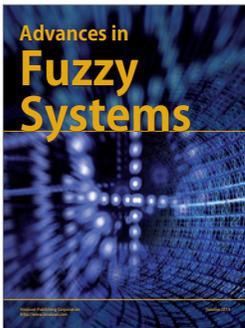
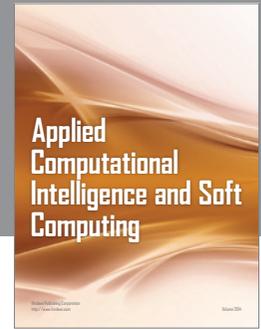
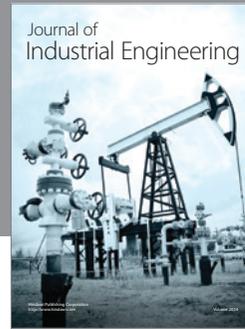
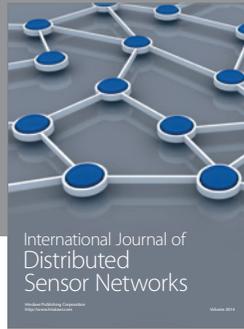
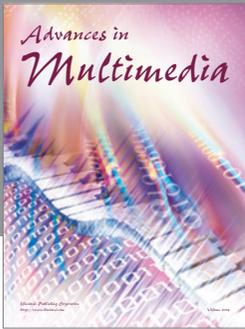
**Yunchuan Sun** received his Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Science, Beijing, China. He now works at the School of Economic and Business Administration, Beijing Normal University, Beijing, China. His current research interests include the theory and application of semantic link network and Internet of Things.

**Hongli Yan** received his B.S. degree in 2012 from Beijing Normal University. He is currently a postgraduate student in the College of Information Science and Technology. His current research interests include the Internet of Things, data mining and computational intelligence.

**Cheng Lu** received her B.S. degree in 2012 from Beijing Normal University. She is now a postgraduate student in the College of Information Science and Technology. Her current research interests include the Internet of Things, Mobile Social Networks and computational intelligence.

**Rong-Fang Bie** received her Ph.D. degree in 1996 from Beijing Normal University, where she is now a professor. She visited the Computer Laboratory at the University of Cambridge in 2003. Her current research interests include knowledge representation and acquisition for the Internet of Things, computational intelligence and model theory.

**Zhangbing Zhou** is an associate professor at the School of Information Engineering, China University of Geosciences (Beijing), and serves as an adjunct associate professor at the computer science department, Institut Mines TELECOM/TELECOM SudParis, France. He received his Ph.D. from the Digital Enterprise Research Institute (DERI), National University of Ireland, Galway (NUIG). His research interests include process-aware information system, service-oriented computing, spatial and temporal database, and sensor network middleware. Email: zhangbing.zhou@gmail.com.



**Hindawi**

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

