# User Interoperability With Heterogeneous IoT Devices Through Transformation

Guangyi Xiao, *Member*, *IEEE*, Jingzhi Guo, *Member*, *IEEE*, Li Da Xu, *Senior Member*, *IEEE*, and Zhiguo Gong, *Member*, *IEEE*

***Abstract*—Heterogeneous device services generated by various devices in different contexts prevent users from efficiently and correctly consuming device services. This seriously hinders the development of Internet of Things. This paper addresses the problems appearing in device discovery and device interaction. It devises a user interoperability framework (UIF) to enable device users to interoperate with heterogeneous devices of different contexts with consistent syntax and semantics. In this framework, a new separation strategy is provided; a device representation method for real, common, and virtual devices is devised; and a device transformability model is proposed to guarantee the proper transformation of device syntax and semantics. To demonstrate the correctness of UIF, a UIF prototype is implemented and several experiment methods are compared to determine which one should be adopted as semantic relatedness computing tools in device discovery for device users and in common device publishing for device providers.**

***Index Terms*—Collaborative sign, cosign, device, device interoperability, Internet of Things (IoT), transformation, user interoperability.**

## I. Introduction

THE BASIC idea of Internet of Things (IoT) [1], [12] is the connectivity of things through Internet protocols, where things are devices such as radio frequency identification (RFID) tags, sensors, actuators, and mobile phones [24], [39]. IoT can be widely applied in many fields [1] such as logistics [21], healthcare [5], and energy [15] in the wide scopes of home, community, city, and country [12]. The extended idea of IoT is, how to discover IoT devices and use the services that the devices provide for various industrial and commercial purposes [1], [10], [40]. To realize these ideas, some key IoT technologies, such as universal IoT device identification, IoT device's semantic information integration, interpretation, and exchange, must be developed [22], [41]. Nevertheless, in developing these technologies, the following challenges are presented.

1) *Large Scale of Co-Operation*: The activities of distributed IoT devices on Internet require cooperation and coordination of thousands or millions of distributed devices.
2) *Global Heterogeneity* [47]: IoT devices and their subnets are highly heterogeneous.
3) *Unknown IoT Device Configuration*: IoT devices from unknown owners have different configuration methods.
4) *Semantic Conflicts*: Different processing logics applied to same IoT networked devices or applications.

One of the reflections of these challenges is the interoperability problem [26], [35], [42] among different *IoT devices* (simply called *device* later in this paper) and between a device and a device user. To illustrate and understand the challenging problem, a motivational example is described as follows.

Suppose that there are two devices A and B, and a device user U (see Fig. 1). Both A and B provide a service of using real-time telescope to view scenery spots through IoT. A's service locations include West Lake of Hangzhou, China, while B's service locations include the Niagara Falls. However, the service designs of A and B are different in both syntax and semantics. User U does not know A and B, but he/she wants to use the real-time telescope services in the scenery spots of West Lake and the Niagara Falls in order to determine which of the two places will be his/her next holiday venue. Since there are no integrated device service catalog, U can compare and select telescope services of A and B only if he/she can find the device service websites of A and B and can understand their service descriptions.

In Fig. 1, in order to explicitly exemplify interoperability problem among A, B, and U, we assume that A and B use Chinese and English to describe device services, respectively. The user U speaks Chinese. It is obvious that User U, on one hand, is difficult to find telescope device, and on the other hand, he/she cannot understand the telescope service provided by B. Similarly, A and B cannot mutually understand each other.

In this paper, we call the interoperability problem [26], [35], [42] that happens between a device user (e.g., U) and a device (e.g., device A or B) as a *user interoperability problem* and call the interoperability problem that happens between two devices (e.g., devices A and B) as a *device interoperability problem*.

In existing researches, some attempt to solve the device interoperability problem [27], [33] and/or user interoperability problem [13]. For example, researches in [10], [23], [27], and [33] propose to build universal middleware for device interoperability in heterogeneous home and enterprise networks. The work in [13] and [28] suggests integrating devices using Web services. Nevertheless, it is still not clear how device users can semantically interoperate with devices without meaning discrepancy.

This paper focuses on resolving the user interoperability problem (note: device interoperability problem will not be discussed in this paper). It has identified that the user interoperability

Manuscript received September 29, 2013; revised December 14, 2013 and January 25, 2014; accepted January 29, 2014. Date of publication February 17, 2014; date of current version May 02, 2014. This work was supported in part by the University of Macau Research Committee under Grant MYRG156-FST11-GJZ. Paper no. TII-13-0687.

G. Xiao, J. Guo, and Z. Gong are with the University of Macau, Taipa, Macau (e-mail: ya97409@umac.mo; jzguo@ umac.mo; fstzgg@umac.mo).

L. D. Xu is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China; with Shanghai Jiao Tong University, Shanghai 200240, China; with the University of Science and Technology of China, Anhui 230026, China; and also with Old Dominion University, Norfolk, VA 23529 USA (e-mail: LXu@odu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2014.2306772

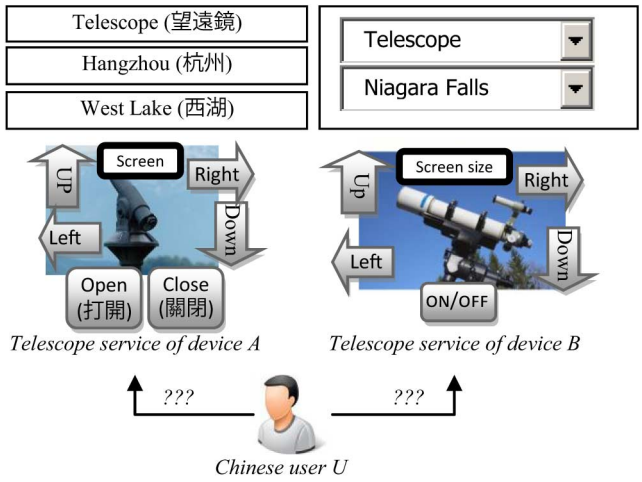


Fig. 1. Motivational example.

problem can be solved if both of the following can be solved: 1) a device discovery problem (the capability of a device user to find a device he/she needs); and 2) a device interaction problem (the capability of a device user to interact with a device he/she has found for device service).

The issue of device discovery relates to the issues of device accessibility and description such that a device must be uniquely accessible and described. Device accessibility is often resolved by promoting IPv6 [6] for connecting to devices and adopts an electronic product code (EPC) scheme [7], [8] to uniquely identify a device. However, problems remain unsolved in device descriptions.

**Problem 1** (*Device categorization*): How existing device services can be cataloged for device users to find.

**Problem 2** (*Device sense disambiguation*): How a new device can be added into an existing device catalog without semantic ambiguity.

Device interaction issue relates to device heterogeneity such that heterogeneous devices (such as A and B) must be transformable to device user's acceptable forms in both syntax and semantics. This further triggers the following problems.

**Problem 3** (*Syntactic device interoperability*): The device instruction format from a user to a device is executable by the device and the message format from the device responding to the user is executable by the user's computer.

**Problem 4** (*Semantic device interoperability*): The meaning of a user's instruction sent from the user to a device is interpretable by the device exactly as the user originally means, and the meaning responding from a device to a user is interpretable by the user exactly as the device originally means.

Problems 3 and 4 are challenging since device users and devices are situated in different contexts, as shown in Fig. 1. Existing solutions to syntactic interoperability often rely on open standards; e.g., standards of Simple Object Access Protocol (SOAP) and Web services for device descriptions [28]. Differently, semantic interoperability of devices is often solved by providing a set of ontologies [22] as common meaning encoding for devices. Unfortunately, not all device providers will adopt same set of ontologies because of their different contexts. This makes device interaction difficult.

This paper aims at proposing a novel user interoperability framework (UIF) enabling device discovery and device interaction, such that any device situated in different contexts can be discovered and interacted by a device user in his/her own context. The central idea of our solution is the strategy of separation and mapping: devices are separated in three device modes of real devices, common devices, and virtual devices, which are respectively provided by real device providers, common device providers, and device users. Real and virtual devices are syntactically and semantically connected for device discovery and interaction through mapping onto a set of common devices. Such a strategy is sufficient to solve the user interoperability problem, i.e., Problems 1–4.

The remainder of this paper is organized as follows. Section II discusses the related work. Section III presents an overview of UIF. Section IV describes a methodology about how various modes of devices are generically represented. Section V proposes a device transformability model. Section VI implements UIF in a UIF prototype. In Section VII, an experiment is conducted on evaluating the accuracy of publishing common devices. Finally, a conclusion is made to summarize the paper with a list of contributions of this paper.

## II. Related Work

In IoT research field, device discovery [39] is often a process of user participation to find the right devices present on Internet directly or indirectly by searching or browsing a device catalog. Existing approaches to device discovery generally adopt unique identifiers of devices and their descriptions to discover devices [39]. Differently, device interaction [33] is a next process of device discovery, in which the user or another device interacts with the device for his/her/its desired services. This section reviews technologies of device discovery and device interaction in IoT area.

### A. Identifiability and Categorization for Device Discovery

Identifiability means that all device objects connected in IoT must be identified for the remote access. It is key to device discovery for a device.

*1) EPC Codes as Unique Identifiers:* In IoT area, EPC codes are an identification standard developed by GS1 AISBL in a uniform resource identifier (URI) form [7], [8], [10]. They are used to identify device objects and capture their static data that map onto relevant products. The advantage of EPC code is its unique identification of a static device object referring to a product. The drawback is that different EPC schemes may encode a same product in heterogeneous EPC codes. This might cause the problem of term sense ambiguities [32] and requires a process of term sense disambiguation [30], [31], [34], [38].

*2) IP Addresses as Unique Identifiers:* In practice, IP for smart objects (IPSO) (www.ipso-alliance.org) promotes IPv6 as the premier solution for accessing to and communicating with smart objects (i.e., devices) including sensors, actuators, and communication devices [6]. However, when a device is in moving, IP address is incapable of providing a unique identification of an object. Even roaming is allowed [6], the device roaming function is based on unique device identity and not IP address. This implies that a device must apply a unique ID instead of IP addresses to combat with device roaming.

*3) URIs as Unique Identifiers:* Guinard and Trifa [13] propose to consider sensor nodes as RESTful resources [36], [37], thus each Internet-based device can be uniquely identified and accessed through URI. Since RESTful is a popular web service application programming interface (API), a device can have a built-in Web server to provide device web service, where URI is used to identify any device. However, devices without web server cannot be identified for the remote access. This implies that a widely acceptable solution of device discovery cannot built on individual device URI scheme, but on a Web server always connectable to both Internet and devices.

*4) UNSPSC and ecl@ss as Device Catalogs:* Assuming any device is uniquely identified and accessible, device users are still impossible to know all device IDs or remember many device IDs for accessing to their needed devices. They usually find devices by using keywords or descriptions in search engines or browsing on device catalogs to discover devices. There are two existing standards on commodity description and classification, which are UNSPSC (www.unspsc.org) and ecl@ss (www.eclass.de). These two standards can be introduced to IoT for device categorization purpose.

### B. Syntactic Interoperability for Device Interaction

In IoT research, syntatic interoperability between devices and device users resolves syntactic heterogeneity to ensure the syntactic consistency of device message formats and message flows. It enables device information to be correctly received, sent, and consumed without communication and use problems. Existing approaches to improving syntactic interoperability are multiple. For example, some popular approaches adopt methods of mash-up devices with service-oriented computing (SOC)-based achitecture [13], Web services [14], RESTful web services [29], open standard protocols [4], and closed protocols [3]. They are all designed for the remote access to devices for information exchange. Untill now, a large number of open and proprietary wireless standards and technologies have appeared for syntactic interoperability, such as the open standards of IEEE 802.15.4 [4], ZigBee [3], WirelessHART [40] as well as proprietary technologies such as Z-Wave [33]. However, these standards are heterogeneous and incompatible with each other. This situation hinders the IoT development.

To solve this problem, middleware technologies are applied in IoT research [10], [23], [27], [33]. For example, home automation network [27], [33] proposes a software universal middleware bridge for interoperability to map dynamically the physical devices in all different middleware domains. Based on the maps of the commands and parameters of the open middlewares on home automation network, one device of a middleware can be discovered and controlled by a system on another middleware. This solution is applaudable for syntactic interoperability. Nevertheless, in reality, not all devices in different contexts know proposed middleware specifications. Thus interoperability remains a problem.

To solve cross-context syntactic interoperability problem, Collaborative Concept Exchange (CONEX) project devises a tree-alike XML syntax called XML Product Map (XPM) [19]. The advantage of XPM syntax is that any message or document is represented as a tree such that the DTD of XPM, shown in Fig. 2, only contains one XML element "sign" with a fixed number of attributes.

```
xpm.dtd
<!ELEMENT sign (#PCDATA | sign)*>
<!ATTLIST sign iid CDATA #IMPLIED
    term CDATA #IMPLIED ctx CDATA #IMPLIED
    dt CDATA #IMPLIED sgt CDATA #IMPLIED>
```

Fig. 2. XPM DTD.

The DTD, shown in Fig. 2, guarantees that XPM files are universally executable in XML syntax when a mini-XPM parser is automatically downloadable to any programs. This syntax makes syntactic interoperability possible among various heterogeneous systems. It avoids the conflicts of ad hoc and heterogeneous XML tag definitions and nesting.

### C. Semantic Interoperability for Device Interaction

Semantic interoperability in IoT research considers a higher-level problem in device message meanings sent and received between devices and/or human users. It requires semantic consistency, i.e., correct meaning interpretation, of device messages among devices, software programs, and human. For example, in Fig. 1, the provider of device A uses "open(打開)/close(關閉)" meaning "turn on and turn off" while the provder of device B uses "ON/OFF." Obviously, these two devices cannot be semantically interoperable for users if one can only understand English or Chinese. This example shows the problem of cross-context semantic interoperability.

Ontology is often used to solve semantic interoperability problem in IoT research [22], [41]. "An ontology is an explicit specification of a conceptualization" [11]. It is a set of objects and relationships among the objects. Applying the concept of ontology, the research in [22] describes devices in three ontologies of a device ontology, a physics domain ontology, and an estimation ontology. These three ontologies are applied to composite services for device discovery for device interaction. Wang *et al.* provided a comprehensive ontology for IoT knowledge representation [41]. Ontology-based device service composition is useful for semantic interoperability when device interoperation is confined in the scope of defined domains. Nevertheless, when various devices are provided in different contexts or domains (as shown in Fig. 1), it is difficult to disambiguate the senses of a same or similar device services. This is because ontologies developed in different contexts cannot guarantee the semantic consistency between heterogeneous device descriptions. Obviously, this is a natural limitation of ontology-based design for resolving semantic interoperability problem because different device providers and users have their own contexts of device designs and uses.

To overcome the limitation of ontology and enable cross-context semantic interoperability for any objects, the collaborative conceptualization theory [18] was developed to define any objects as collaborative concepts. These concepts represent things (including devices) and/or nonthings, as *signs* of semiotics. It assumes that any sign is contextual and inherently semantically heterogeneous when signs were born in different contexts. Thus, when any people or system agents from different contexts intend to unambiguously talk with each other, they must first collaborate to build semantically consistent signs. Signs of

vocabularies designed based on such a theory are semantically consistent and can be applied in different cross-context application designs (e.g., [20]).

A collaborative sign (cosign) [16] is generically represented as follows:

$$\text{cosign} = (S, C, X, \diamondsuit)$$

where $S$ is a cosign internal identifier IID (also called structure), $C$ is a natural language term denoting $S$, $X$ is the context of $S$, and $\diamondsuit$ is the definition interpreted by someone (called interpretant). The process of making a cosign to a *cosign dictionary* is as follows.

1) Given a cosign collaborative editing system (e.g., [16], [17]) where sign collaborators are joined.
2) Collaborator $X$ proposes a cosign, e.g., telescope, together with its definition $\diamondsuit$, e.g., "a telescope is a long instrument shaped like a tube. It has lenses inside it that make distant things seem larger and nearer when you look through it." Other collaborators other than $X$ discuss, revise, and finally make an agreement on the definition of the cosign "telescope."
3) A unique cosign internal identifier IID $\in S$, e.g., 1111, is assigned to the cosign "telescope" and its definition.
4) When a user uses cosigns, a term is looked up in the cosign dictionary. If it exists, it will be used directly otherwise steps 2) and 3) are launched.

In the rest of this paper, cosigns will be adopted as the atomic building bricks for device syntax and semantics.

## III. Overview of UIF

To enable user interoperability with devices across heterogeneous contexts, this section proposes a novel UIF, shown in Fig. 3. UIF takes a separation strategy such that devices have three presentation modes of real device, common device, and virtual device. Each device mode is separated in an autonomous scope but transformable into another device mode of another scope.

By such a separation strategy, we guarantee that there is no need for device providers to remake devices in order to deliver their device services to contextually different device users. For example, providers of devices A and B, shown in Fig. 1, can continue to use their original device services for device service provision without modification.

### A. Device Scopes

Devices are defined by different purposes in their own scopes. A *real device scope* is a class of contexts consisting of a set of *real device providers* (called "device provider") (e.g., providers of device A and B in Fig. 1). Each device provider has its own context and has a set of real devices that provides device services. In general, service specifications (i.e., syntax) and service content descriptions (i.e., semantics) are context-dependent on different device providers. A *common device scope* is a common context, consisting of a set of collaborative cosign providers (called "cosign provider"), a set of cosigns as a commonly understandable dictionary, a set of common devices, a common device catalog (CDC), and a mapping editor. A *virtual device scope* is a class of contexts consisting of a set of *device users* (e.g., user U in Fig. 1). Each device user has a stored Web page (containing a tool of *device finder*) and a set of virtual devices found by device finder.

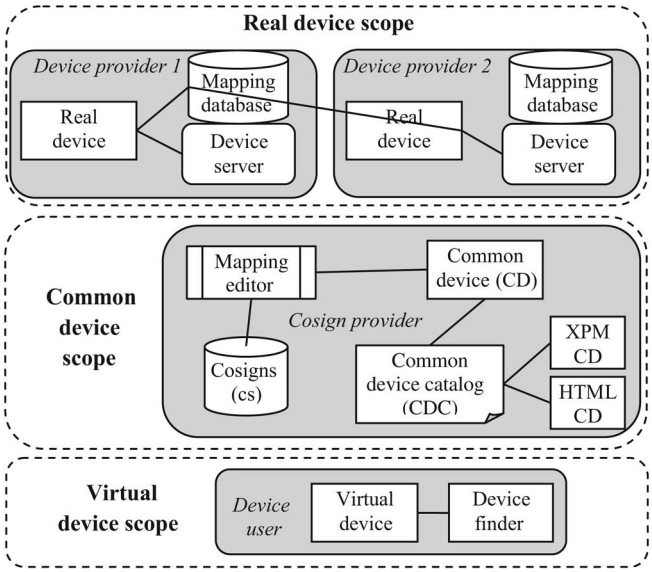


Fig. 3. User interoperability framework.

Separating device modes in different scopes is helpful for both device providers and device users. Device providers can retain the legacy designs of real devices and device services without requiring redesigning existing real devices and service methods for device service provision. Device users, on the other hand, does not need to know device providers before using device services and can keep their own preferences on using device services.

### B. UIF Role Responsibilities and Expected Outcomes

In UIF framework, device providers, cosign providers, and device users have different responsibilities. A *device provider* is responsible for creating any common device based on its local real device and publishes it to CDC. Its responsibility and expected outcomes could be described in the procedure of Fig. 4.

In Fig. 4, a device provider uses a mapping editor to map a real device onto a common device by using cosigns either in the form of XPM syntax (see Section II) (which is supposed to be used by other device providers) or in the form HTML syntax (which is supposed to be used by device users). It stores the mapping file between real device and common device in its own mapping database for later use and publishes the common device to CDC. It must also check whether there exists a same device already published in CDC through a *sense disambiguation mechanism* (SDM) procedure [20]. A device provider can also act as a user of common device (in XPM format). It can receive an XPM common device and retrieve part or all device information to redesign its local real device as a new common device and submit to CDC for providing real device services.

*Cosign providers* are responsible for creating cosigns (i.e., a dictionary of collaborative terms) and maintaining a CDC by collaboration on a collaborative editing system. The CDC is

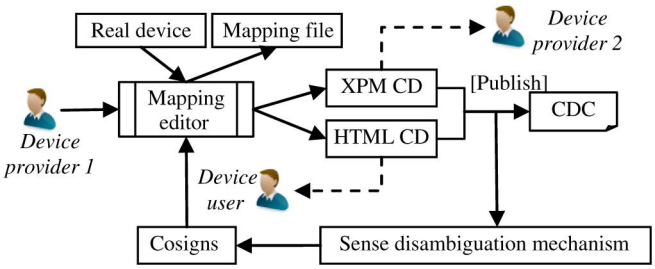


Fig. 4. Real device provider's responsibility.

created and encoded based on the existing electronic product standard UNSPSC (unspsc.org) by adding the mappings between UNSPSC classifiers and cosign IIDs. The particular design method of cosign dictionary and CDC is out of the research scope of this paper (see the researches of [17] and [18]).

A *device user* is responsible for locating a common device (in HTML format) through the tool of device finder, which will convert a common device into a user-understandable virtual device for device user to use real device services.

### C. Chain of Device Transformation

In essence, the general idea of UIF framework to solve user interoperability problem is to transform a real device into a common device and further into a virtual device, and vice versa, along a *chain of device transformation*, such that:

1) virtual device → common device → real device
2) real device → common device → virtual device

where "→" denotes a transformation in both syntax and semantics such that real device, common device, and virtual device are understandable in the scopes of real devices, common devices, and virtual devices, respectively.

Apparently, the transformation sequence guarantees that 1) the instructions of device user to use a device can arrive at real device without ambiguity; and 2) real device service can be sent to device user from device provider without ambiguity.

Overall speaking, the provision of CDC solves Problem 1 of device categorization, the provision of a SDM solves Problem 2 of device sense disambiguation, and the provision of the chain of device transformation solves Problems 3 and 4 of syntactic and semantic device interoperability problems.

In Section IV, we describe how three modes of devices are actually represented in different device scopes.

## IV. Device Representation Methodology

To enable a mode of a device to be successfully transformed into another mode of the same device, we first model any device in a generic representation. Since any device can be abstracted as an object with a set of controls, each control consists of a set of commands sent from a device user and a response to the commands sent back from the device. This abstraction depicts a device as a user interaction between a user and a device, consisting of a set of controls with a set of commands sent from a device user to a device and a set of command responses from the device. Formally, we generalize a device as a composite sign [16] as follows.

*Definition 1 (Generic Device Representation):* A device is a composite interactive sign $D^{\Leftrightarrow}$, such that

$$D^{\Leftrightarrow} ::= \text{sign}(\text{sign}^{+}(\text{sign}^{+}))$$

where $D^{\Leftrightarrow} ::= D^{\Rightarrow} + D^{\Leftarrow}$, in which $D^{\Rightarrow}$ represents any sign for a device carrying device user's control commands sending to a device and $D^{\Leftarrow}$ represents any control command's response of a device sent back to the user. "+" means one or more signs. In this definition, the first level sign denotes a device, the second level sign refers to a device control, and the third level sign denotes either a device control command or a device control command response. Generically, a sign representation is a tuple as follows:

$$\text{sign} ::= \text{sign}(\text{ID}, N, D, P, U, L, S, T, V)$$

where ID is the unique identifier to denote unique device ID, control ID, or command ID; N is the name of device, control, or command; $D$ is the definition of a name $N$; $P$ is a device provider; $U$ is a device user; $L$ is a natural language used to describe the name $N$ and the definition $D$; $S$ is the sign type of signs; and $T$ is the data type constraining values $V$.

Based on Definition 1, any real device, common device, and virtual devices can be further represented.

### A. Real Device Representation

*Definition 2 (Real Device Representation "$D_r^{\Leftrightarrow}$"):*

$$D_r^{\Leftrightarrow} ::= \text{sign}[N_r, D_r, P_r, U_r, S_r] \times (\text{sign}^{+}[N_r, D_r, S_r](\text{sign}^{+}[N_r, D_r, S_r, T_r, V_r])).$$

For example, the device interaction of the telescope $B$ with the device user $U$ in Fig. 1 can be represented in Fig. 5 to model the real device of telescope $B$.

### B. Common Device Representation

A *common device* is used to integrate various heterogeneous real devices supplied by contextually different device providers. It is understandable by device user and other programs when being properly interpreted. Technically, it aims at providing a universally understandable device representation syntax and semantics (i.e., meaning) to device providers, device users, and other programs. Strategically, any common device representation is designed and provided by device providers following HTML/XPM syntax and cosign dictionaries, discussed in Section II.

*Definition 3 (Common Device Representation "$D_c^{\Leftrightarrow}$"):*

$$D_c^{\Leftrightarrow} ::= \text{sign}[\text{ID}_c, \text{ID}_P, \text{ID}_U, S_c] \times (\text{sign}^{+}[\text{ID}_c, S_c](\text{sign}^{+}[\text{ID}_c, S_c, T_c, V_c]))$$

where $D_c$ is a set of common devices $\in$ cosigns and $\text{ID}_c$, $\text{ID}_P$, $\text{ID}_U \in \text{IID} \in$ cosigns. Definition 3 can be implemented in two forms of XPM common device representation for device-to-device interaction and HTML common device representation for user-to-device interaction.

*XPM common device representation* is based on XPM syntax. Applying XPM syntax, a common device is represented as an XPM common device command file template (XPMCDC) and

**From any user (command file):**

$D_r^{\Rightarrow}$ ::= sign[$N_r$="telescope", $P_r$="Device provider B", $S_r$="device"](
sign[$N_r$="screen", $S_r$="control"](
sign[$N_r$="size", $S_r$="command", $T_r$="2int", $V_r$="1200, 800"])
sign[$N_r$="switch", $S_r$="control"](
sign[$N_r$="on", $S_r$="command", $T_r$="bool", $V_r$="true"])
……)

**From the device (response file):**

$D_r^{\Leftarrow}$ ::= sign[$N_r$="telescope", $U_r$="Device user U", $S_r$="device"](
sign[$N_r$="screen", $S_r$="control"](
sign[$N_r$="size", $S_r$="response", $T_r$="bool", $V_r$="true"])
sign[$N_r$="switch", $S_r$="control"](
sign[$N_r$="on", $S_r$="response", $T_r$="bool", $V_r$="true"])
……)

Fig. 5. Real device example for telescope from Fig. 1.

```
<!DOCTYPE sign SYSTEM "xpm.dtd">
<sign iid="1111 (telescope id)" ctx="8888 (device provider's id)"
                                    sgt="iot:device">
   <sign iid="11112 (switch id)" sgt="iot:device:control">
      <sign iid="111121 (on id)" sgt="iot:device:control:command">
         <sign iid="111121 (on id)" sgt="iot:device:control:command:value"
                                     dt="data type">true (a value)</sign>
      </sign> </sign></sign>
```

Fig. 6. XPM common device command representation.

```
<!DOCTYPE sign SYSTEM "xpm.dtd">
<sign iid="1111 (telescope id)" ctx="8888 (device provider's id)"
                                    sgt="iot:device">
   <sign iid="11112 (switch id)" sgt="iot:device:control">
      <sign iid="111121 (on id)" sgt="iot:device:control:response">
         <sign iid="111121 (on id)" sgt="iot:device:control:response:value"
               dt="data type">9000 (a value for success)</sign>
      </sign> </sign></sign>
```

Fig. 7. XPM common device response representation.

an XPM common device response file template (XPMCDR), following Definition 3.

*Definition 4 (XPMCDC):* XPMCDC is an XPM common device representation for a device with a set of controls and a set of control commands, as shown in Fig. 6.

In Fig. 6, iid = "1111" $\in \text{sign}[\text{ID}_c]$, ctx = "8888" $\in \text{ID}_P$, iid = "11112" $\in \text{sign}(\text{sign}[\text{ID}_c])$, iid = "111121" $\in \text{sign}(\text{sign}(\text{sign}[\text{ID}_c]))$, sgt = "…" $\in S_c$, dt = "…" $\in T_c = \text{iot}:\text{device}|\text{iot}:\text{device}:\text{control}|\text{iot}:\text{device}:\text{command}|\text{iot}:\text{device}:\text{command}:\text{value}$, and the value of a $\text{sign} \in V_c$.

*Definition 5 (XPMCDR):* XPMCDR is an XPM common device representation for a device with a set of controls and a set of control command responses, as shown in Fig. 7.

In Fig. 7, each response corresponds to a command of a control.

In Fig. 6, a device user triggers a "switch" control with command "ON" for "ON" to be set as "true." In Fig. 7, the device responds the user with a code value "9000."

*HTML common device representation* is based on HTML syntax. It is suitable for device users of virtual devices to derive virtual devices with the following use pattern.

1) Search a device catalog and find his/her needed device.
2) Download the device automatically as an HTML page.

```
<html><body><fieldset id="1111" style="width: 200px"><legend>&cs#1111;</legend>
<!--&cs#1111; = telescope -->
<label id="8888">&cs#8888;</label> <!--&cs#8888; = Corporation B -->
<form action="urlB2"method="post" id="11112" name="&cs#11112">
<!--&cs#11112; = switch -->
      &cs#111121;:<input id="111121" type="checkbox" checked="checked"/><br/>
      <!--&cs#111121; = on -->
      <input id="333333:2" type="submit" value="&cs#333333:2;"/></form>
      <!--&cs#333333:2; = update -->
</fieldset></body></html>
```

Fig. 8. HTML representation of a common device (with a swith control).

```
<html lang="en"><body>
<fieldset id="1111" style="width: 200px"><legend>telescope</legend>
<label id="8888">B Corporation</label>
<form action="urlB2" method="post" id="11112" name="switch">
      On:<input id="111121" name="on" type="checkbox" checked="checked"/><br/>
      <input id="333333:2" type="submit" value="update"/></form>
</fieldset>
</body></html>
```

Fig. 9. HTML representation of a virtual telescope.

3) Fill in the use instructions (i.e., control commands) as an HTTP request on the device page.
4) Device server sends back the device service (i.e., control command responses) as an HTTP response.

In this paper, we propose an HTML representation of common device, shown in Fig. 8. In this proposal, an HTML group box (i.e., <fieldset>) represents a common device $\text{sign}[\text{ID}_c]$ (e.g., telescope), a form (i.e., <form>) represents a control $\text{sign}(\text{sign}[\text{ID}_c])$ (e.g., switch), HTML controls such as <input> (e.g., "on" and "update") represent commands $\text{sign}(\text{sign}(\text{sign}[\text{ID}_c]))$, and <label> represents a context ctx of $\text{ID}_P$ or $\text{ID}_U$ (e.g., Corporation B).

The graphic display of Fig. 8 is a common device with a switch control. This control consists of a checkbox command with the meaning of "ON" to open telescope and a command with the meaning of "update" to submit a request.

The novelty of both XPM and HTML representations on common devices is not the use of XPM or HTML syntax, but a natural language-neutral method of using IID of cosigns to replace all natural language-dependent content in a common device file such as device name, control name, command name, or a device provider's name. By this method, any common device file is not only syntactically interoperable but also semantically interoperable. The receivers of a common device file only need to display the IID of cosigns in their locally used natural languages, the common device file is then semantically interoperable with device users.

### C. Virtual Device Representation

A *virtual device* is a personal device used to consume real device services as if device users are using real devices. It requires that the service contents from the real devices are understandable by device users. At least device users know how to control a remote real device on a virtual device.

*Definition 6 (Virtual Device Representation "$D_v^{\Leftrightarrow}$"):*

$$D_v^{\Leftrightarrow} ::= \text{sign}[\text{ID}_v, N_v, P_v, L_v, S_v] \times (\text{sign}^+[\text{ID}_v, N_v, S_v](\text{sign}^+[\text{ID}_v, N_v, S_v, T_v, V_v])).$$

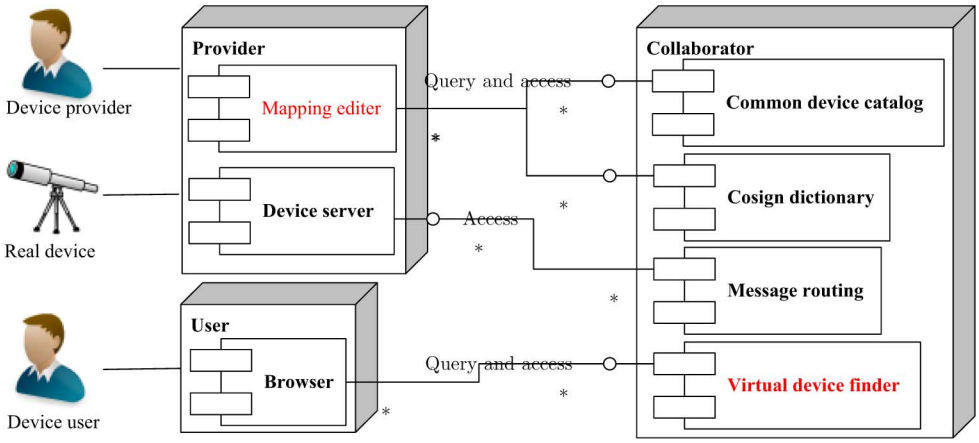


Fig. 10. Structure of UIF Prototype.

A virtual device representation is always in HTML format transformed by device user based on HTML common device representation. It replaces the cosign IIDs of a common device representation with readable terms for the names of device, controls, commands, etc. Fig. 9 shows an example of the virtual device for telescope transformed from Fig. 8.

In Fig. 9, the cosign iids are transformed as English terms such as "&cs#1111;= telescope," "&cs#8888;= Corporation B," "&cs#11112;= switch," "&cs#111121;= on," and "&cs#333333:2;= update." A virtual device in this sense is a personalized representation of a common device in a user's context.

## V. Device Transformability Model

The service of a real device is available and usable to a device user only when the service can be delivered to the virtual device in device user's manner. This section describes a device transformability model based on the concept of chain of device transformation provided in Section III-C.

*Definition 7 (Real-Common Device Mapping):* Given a real device $D_r$ of Definition 2 and a common device $D_c$ of Definition 3, $\mathrm{map}(D_r, D_c)$ exists if and only if there exists:

1) $\mathrm{map}(\mathrm{sign}[N_r], \mathrm{sign}[\mathrm{id}_c])$ for two device mapping;
2) $\mathrm{map}(\mathrm{sign}[N_r] \cdot [N_r], \mathrm{sign}[\mathrm{id}_c] \cdot [\mathrm{id}_c])$ for all control mappings;
3) $\mathrm{map}(\mathrm{sign}[N_r] \cdot [N_r] \cdot [N_r], \mathrm{sign}[\mathrm{id}_c] \cdot [\mathrm{id}_c] \cdot [\mathrm{id}_c])$, for all command mappings;
4) $\mathrm{map}(\mathrm{sign}[N_r] \cdot [N_r] \cdot [N_r] \cdot t_r : v_r, \mathrm{sign}[\mathrm{id}_c] \cdot [\mathrm{id}_c] \cdot [\mathrm{id}_c] \cdot t_c : v_c)$ for all control command or response value mappings.

Since the real device terms and the common device IID appeared in map 1) to 4) of Definition 7 exist and have already been mapped in the cosign dictionary, Definition 7 exists.

*Definition 8 (Common-Virtual Device Mapping):* Given a common device $D_c$ of Definition 3 and a virtual device $D_v$ of Definition 6, $\mathrm{map}(D_c, D_v)$ exists if and only if there exists:

1) $\mathrm{map}(\mathrm{sign}[\mathrm{id}_c], \mathrm{sign}[\mathrm{id}_v])$ for device mapping;
2) $\mathrm{map}(\mathrm{sign}[\mathrm{id}_c] \cdot [\mathrm{id}_c], \mathrm{sign}[\mathrm{id}_v] \cdot [\mathrm{id}_v])$ for all control mappings;
3) $\mathrm{map}(\mathrm{sign}[\mathrm{id}_c] \cdot [\mathrm{id}_c] \cdot [\mathrm{id}_c], \mathrm{sign}[\mathrm{id}_v] \cdot [\mathrm{id}_v] \cdot [\mathrm{id}_v])$ for all control command mappings;
4) $\mathrm{map}(\mathrm{sign}[\mathrm{id}_c] \cdot [\mathrm{id}_c] \cdot [\mathrm{id}_c] \cdot t_c : v_c, \mathrm{sign}[\mathrm{id}_v] \cdot [\mathrm{id}_v] \cdot [\mathrm{id}_v] \cdot t_v : v_v)$ for all control command or response value mappings.

Since $\mathrm{id}_c = \mathrm{id}_v$ for any device, and mappings of controls, commands, and values of the device exist, Definition 8 exists.

*Definition 9 (Device Transformability Model):* Given a real device $D_r$, a common device $D_c$, and a virtual device $D_v$, the $D_r$, $D_c$, and $D_v$ are mutually transformable if and only if:

1) $\mathrm{map}(D_r, D_c)$ exists;
2) $\mathrm{map}(D_c, D_v)$ exists.

The device transformability model described in Definition 9 ensures in theory that any real device is transformable to a virtual device, and vice versa.

## VI. UIF Prototype Implementation

This section will implement a UIF Prototype, shown in Fig. 10, to demonstrate the correctness of the designed approach. This Prototype consists of three nodes of a Provider, a User and a Collaborator.

In Fig. 10, Provider node consists of components of a mapping editor, shown in Fig. 11, for designing and publishing common devices to a CDC (shown in the left pane of Fig. 11) and a Device Server for providing device services and transformation.

Collaborator node consists of components of Virtual Device Finder, Cosign Dictionary, CDC, and Message Routing. In this node, Cosign Dictionary (shown in the right pane of Fig. 11) is a set of signs collaboratively designed by sign designers and commonly understandable to all. CDC is a catalog storing publicly accessible common devices in both formats of XPM and HTML. Message Routing is a tool forwarding messages between nodes of User and Providers. Device Finder, shown in Fig. 12, is a tool helping device users to find real devices and services and create personalized content to device users. User node only consists of a component of Browser for device users to access to Internet by using Device Finder. The use result is the found virtual device, shown in Fig. 13, in which a device user can interact with a real device to consume desired device service.

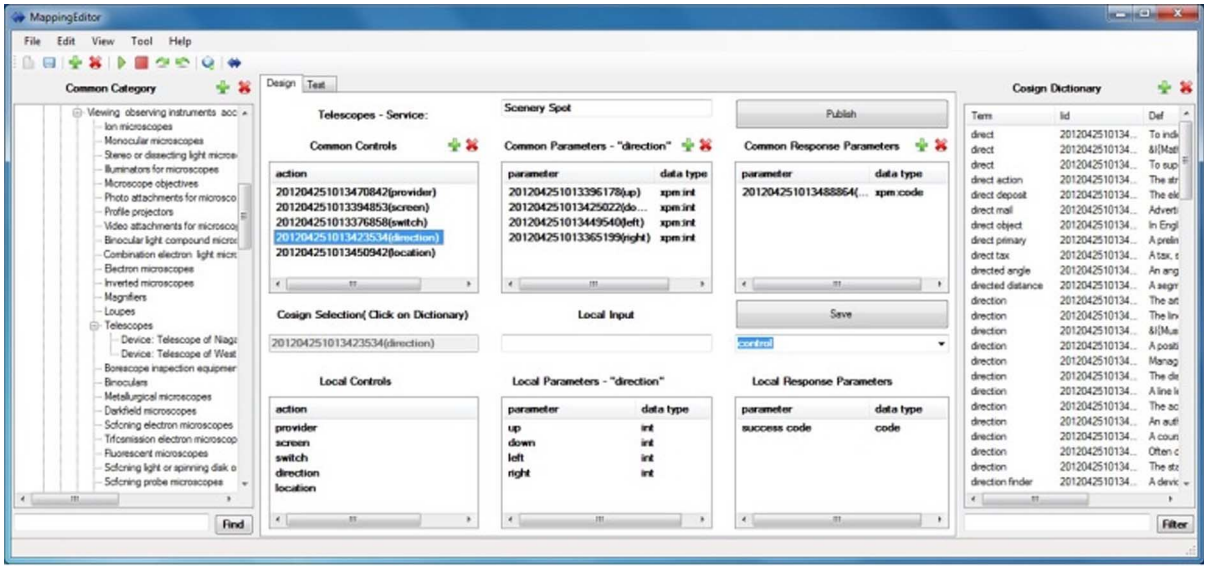


Fig. 11. Mapping editor.

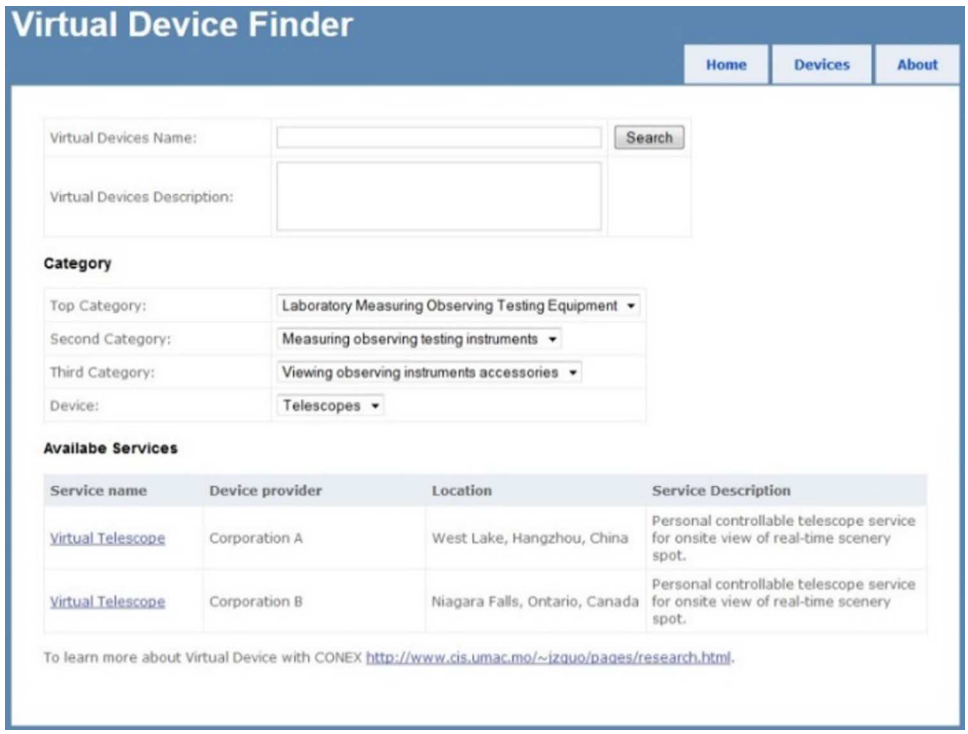


Fig. 12. Virtual device finder server page.

## VII. Experiments and Evaluation

This section makes an experiment on evaluating whether a common device is publishing to a semantically proper category of CDC. Technically, the experiment is to semantically compare the meaning of the category definition of a submitting term (e.g., "telescope" definition in cosign dictionary or device user definition) with the meanings of existing CDC category definitions (e.g., the definitions of "telescopes" category, etc.). Apparently, we expect to find the comparison result between a term definition $T_{\text{new}}$ in cosign dictionary (or a user-provided keyword) and a set of term definitions $T$ in CDC, as follows:

$$\text{Compare}(T_{\text{new}}, T) = \begin{cases} \text{sim} \geq \text{Th} \\ \text{sim} < \text{Th} \end{cases} \quad (1)$$

where sim is the semantic similarity (also called *semantic relatedness*) between the two definitions, and Th is a threshold to determine whether sim is acceptable for publishing mechanism to publish the common device or retrieving device list for device users.

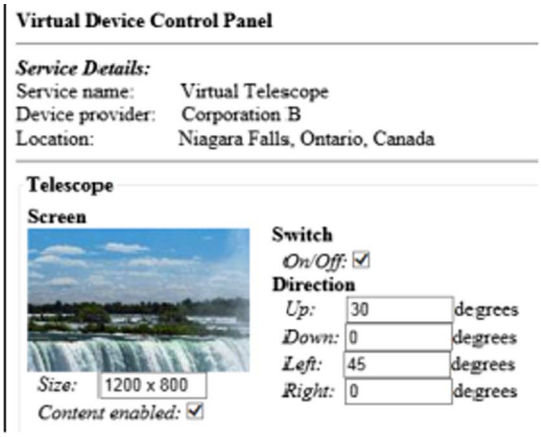


Fig. 13. Virtual device control server page.

TABLE I
EXPERIMENT METHODS

| Method of *Compare*($C_{new}$, $C$) | Method Type |
|---|---|
| Lesk [32] | – |
| TermVector [2] | – |
| WikiESA [9] | Corpus-based |
| WordNetESA | Corpus-based |
| STS2008[25] | Corpus-based |
| SR2T_LCA | Knowledge-based |
| SR2T_WUP | Knowledge-based |
| SR2T_RES | Knowledge-based |
| SR2T_LIN | Knowledge-based |

### A. Experiment Setting

To make the above required experiment, a dataset $D$ for evaluation comes from the research work of evaluating semantic relatedness between any two terms [20]. $D$ is a tuple of ($T$, $S$), where $T$ is a set of items in dictionary, and $S$ is a set of synsets for the items. Each $t_i$ in $T$ is a tuple of ($w_i$, $d_i$). Each $s_j$ in $S$ is a subset of $T$ denoted as $T_{s_j}$, where $T_{s_j} \in T$ and all $t_i$ in $T_{s_j}$ are a synset. Each $t_i$ can only find one $s_j$ by following function $s_j = \text{fs}(t_i, S)$. $D$ is computed from a validation set finding (VSF) algorithm of [20] by computing the corpus of a bilingual dictionary, WordNet, and CILIN. The count of $T$ is $n = 11\,260$, and the count of $S$ is $m = 4236$. Particularly, given $t_x = (w_x, d_x) =$ device providers' common device (device name, device definition) $\in$ cosign dictionary, $t_y = (w_y, d_y) =$ device users' submitted device search (device, device description), and $t_z = (w_z, d_z) =$ category name and category definition (category, category definition), $t_x$, $t_y$, $t_z \subset T$ as a simulation of term and term definition generation. The specific experiments are made on a computer with processor Inter Core i7 central processing unit (CPU) 870 @ 2.93 GHz, memory 4.0 G, and 32-bit Operating System.

### B. Experiment Methods

We set the experiment to find out sim values Th (1) by employing the existing semantic relatedness methods developed in the research of [20], shown in Table I.

### C. Evaluation Criteria

There are two evaluation criteria, which are time cost and accuracy rate in the experiments. *Time cost* is the total computing time of the semantic relatedness computing procedure, shown in Fig. 14. *Accuracy rate* (AR) is the comparison result sim between term definitions. It requires to be high.

$$\text{AR} = |T_2|/|T| \tag{2}$$

where $T$ and $T_2$ are defined in Fig. 14.

**$n \times n$ Computing Procedure**. Identifying a set of accurate queries
**Input**: $T$, $S$
**Output**: $T_2 \leftarrow \emptyset$
1. for($t_i \in T$){
2. $t_{max} \leftarrow \emptyset$, $v_{max} = 0$ /* initialize item max and value max */
3. for($t_k \in T$){
4. $v \leftarrow SRQT(t_i, t_j)$ /* SRQT is a comparison operation */
5. if $v_{max} > v$ then {
6. $t_{max} \leftarrow t_j$
7. $v_{max} \leftarrow v$
8. } /* Swap the maximu item with bigger value of semantic relatedness */
9. } /* Found a maximu item with maximu value */
10. $s_j = fs(t_i, S)$
11. if $t_{max} \in T_{s_j}$ then /* if the $t_{max}$ in the synset of query item $t_i$ */
12. $T_2 \leftarrow T_2 \cup t_{max}$ /* union $T_2$ with $t_{max}$ */
13.}

Fig. 14. Semantic relatedness computing procedure.

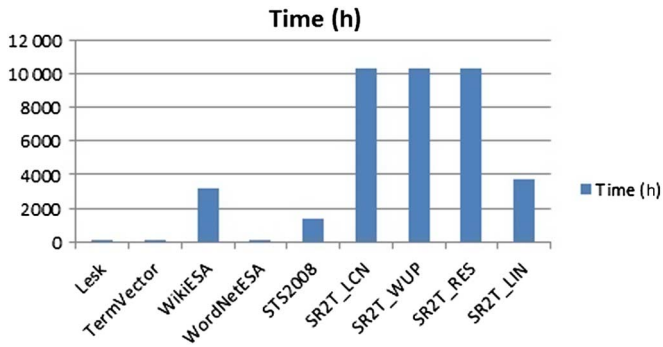


Fig. 15. Time cost of different semantic relatedness methods.

## VIII. RESULT AND DISCUSSION

### A. Time Cost

The time cost of nine semantic relatedness methods for experiments procedure is shown in Fig. 15. Since all knowledge-based semantic relatedness methods such as SR2T_LCA, SR2T_WUP, SR2T_RES, and SR2T_LIN and a corpus-based method such as WikiESA for the query experiment are not efficient in the experiments. They are not suggested to be adopted for the query of semantic relatedness.

### B. Accuracy Rate

After filtering out the semantic relatedness methods of inefficiency, four methods of Lesk, TermVector, WordNetESA, and STS2008 are retained for further analysis. Fig. 16 shows the accuracy rates of the four methods, among which the accuracy rates of Lesk and TermVector are less than 80%.

By analyzing the experimental result of Fig. 16, it is suggested to adopt the approaches of WordNetECA and STS2008 as the tools for semantic relatedness query because they have good

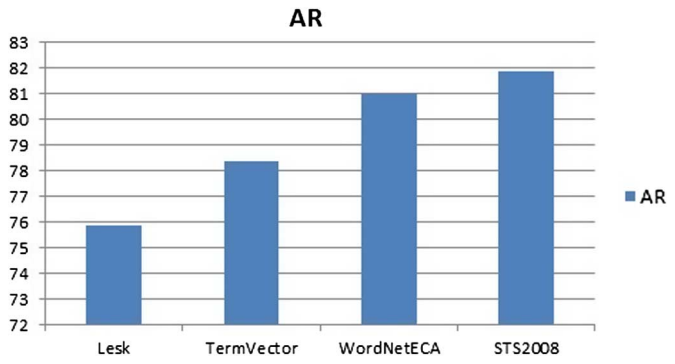


Fig. 16. Precision of semantic relatedness with evaluation along NSG [20].

trade-off of accuracy rate and performance for large scale queries. This suggestion has been implemented in our design of common device publishing mechanism.

## IX. Conclusion and Future Work

This paper has resolved a cross-context user interoperability problem of providing services of heterogeneous IoT devices to Internet-based device users in various syntactic and semantic contexts. It has proposed a novel UIF Framework to enable sets of heterogeneous real devices to be transformable to a set of common devices and finally to sets of virtual devices that are personalized to device users' semantic contexts. The UIF framework adopts a separation strategy where heterogeneous devices are separated in different device modes of real device, common device, and virtual device. This separation allows no need of modifying the services of existing real devices. The core to this strategy is a bi-directional transformation chain between real devices, common devices, and virtual devices, where devices in different forms are mapped for both syntactic and semantic transformation.

The paper has several contributions as follows.

1) It has described a new UIF to solve the interoperability problem between a device user of a context and a device of another context.
2) It has provided a novel device representation method, which enables a same device working in different contexts.
3) It has proposed a useful device transformability model to guarantee that any device is both syntactically and semantically transformable between their representations of different contexts.

In addition, the research result of XPM common device representation can be applied to many other applications. For example, any systems can use XPM common devices to design new devices. This will provide device interoperability between devices. In future, we will plan to launch a third-party platform to accommodate a CDC, where device providers can submit their local real devices as common devices for device users to consume as personalized virtual devices.

## References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval.* Reading, MA, USA: Addison-Wesley, 1999.
[3] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards," *Comput. Commun.*, vol. 30, no. 7, pp. 1655–1695, May 2007.
[4] E. Callaway, P. Gorday, L. Hester, J. A. Gutierrez, M. Naeve, B. Heile, and V. Bahl, "Home networking with IEEE 802.15.4: A developing standard for low-rate wireless personal area networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 70–77, Aug. 2002.
[5] M. Domingo, "An overview of the Internet of Things for people with disabilities," *J. Netw. Comput. Appl.*, vol. 35, pp. 584–596, 2012.
[6] A. Dunkels and J. Vasseur, Ip for smart objects aliance, in *Internet Protocol for Smart Objcts (IPSO) Alliance White Paper*, no. 2, 2008 [Online]. Available: http://www.sics.se/~adam/dunkels08ipso.pdf
[7] *EPC Tag Data Standard*, GS1 Standard Version 1.8, 2014.
[8] EPCglobal Inc., *EPC Information Services (EPCIS) Version 1.0.1 Specification*, Errata Approved by TSC, Sep. 2007.
[9] E. Gabrilovich and S. Markovitch, "Wikipedia-based semantic interpretation for natural language processing," *J. Artif. Intell. Res.*, vol. 34, no. 2, p. 443, Mar. 2009.
[10] K. Gama, L. Touseau, and D. Donsez, "Combining heterogeneous service technologies for building an Internet of Things middleware," *Comput. Commun.*, vol. 35, no. 4, pp. 405–417, Feb. 2012.
[11] T. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, Apr. 1993.
[12] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.
[13] D. Guinard and V. Trifa, "Towards the web of things: Web mashups for embedded devices," in *Proc. Int. World Wide Web Conf.,* Madrid, Spain, Apr. 2009.
[14] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *IEEE Trans. Serv. Comput.*, vol. 3, no. 3, pp. 223–235, Feb. 2010.
[15] V. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. Hancke, "A survey on smart grid potential applications and communication requirements," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 28–42, Feb. 2013.
[16] J. Guo, "Sign-based semantics-enriched social computing," in *Proc. 12th IEEE Conf. Commerce Enterp. Comput.,* Shanghai, China, Nov. 2010, pp. 188–195.
[17] J. Guo, I. H. Lam, C. Chan, and G. Xiao, "Collaboratively maintaining semantic consistency of heterogeneous concepts towards a common concept set," in *Proc. EICS 2010*. Berlin, Germany, 2010, pp. 213–218.
[18] J. Guo, "Collaborative conceptualisation: Towards a conceptual foundation of interoperable electronic product catalogue system design," *Enterp. Inf. Syst.*, vol. 3, no. 1, pp. 59–94, Feb. 2009.
[19] J. Guo, *Collaborative Concept Exchange.* Saarbrücken, Germany: VDM Publishing, May 2008.
[20] J. Guo, L. Da Xu, G. Xiao, and Z. Gong, "Improving multilingual semantic interoperation in cross-organizational enterprise systems through concept disambiguation," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 647–658, Aug. 2012.
[21] Z. Guo, Z. Zhang, and W. Li, "Establishment of intelligent identification management platform in railway logistics system by means of the Internet of Things," *Procedia Eng.*, vol. 29, pp. 726–730, 2012.
[22] S. Hachem, T. Teixeira, and V. Issarny, "Ontologies for the Internet of Things," in *Proc. 8th Middleware Doctoral Symp.,* Lisbon, Portugal, Dec. 2011, article no. 3.
[23] W. He and L. Xu, "Integration of distributed enterprise applications: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 35–42, Feb. 2014.
[24] J. Heo, J. Hong, and Y. Cho, "EARQ: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks," *IEEE Trans. Ind. Informat.*, vol. 5, no. 1, pp. 3–11, Feb. 2009.
[25] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Trans. Knowl. Discovery Data*, vol. 2, no. 2, p. 10, Jul. 2008.
[26] R. Jardim-Goncalves, A. Grilo, C. Agostinho, F. Lampathaki, and Y. Charalabidis, "Systematisation of interoperability body of knowledge: The foundation for enterprise interoperability as a science," *Enterp. Inf. Syst.*, vol. 7, no. 1, pp. 7–32, Feb. 2013.
[27] D. Kim, C.-E. Lee, J. H. Park, K. Moon, and K. Lim, "Scalable message translation mechanism for the environment of heterogeneous middleware," *IEEE Trans. Consum. Electron.*, vol. 53, no. 1, pp. 108–113, Feb. 2007.

[28] R. Kyusakov, J. Eliasson, J. Delsing, J. Deventer, and J. Gustafsson, "Integration of wireless sensor and actuator nodes with it infrastructure using service-oriented architecture," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 43–51, Feb. 2013.
[29] M. Lanthaler and C. Gütl, "On using json-ld to create evolvable restful services," in *Proc. 3rd Int. Workshop RESTful Des.,* Apr. 2012, pp. 25–32.
[30] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," in *WordNet: An Electronic Lexical Database*, C. Fellbaum, Ed., Cambridge, MA, USA: MIT Press, 1998, pp. 265–283.
[31] D. Lin, "An information-theoretic definition of similarity," in *Proc. 15th Int. Conf. Mach. Learn.,* Madison, WI, USA, 1998, pp. 296–304.
[32] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone," in *Proc. SIGDOC'86,* Toronto, ON, Canada, 1986, pp. 24–26.
[33] K.-D. Moon, Y.-H. Lee, C.-E. Lee, and Y.-S. Son, "Design of a universal middleware bridge for device interoperability in heterogeneous home network middleware," *IEEE Trans. Consum. Electron.*, vol. 51, no. 1, pp. 314–318, Feb. 2005.
[34] M. Palmer and Z. Wu, "Verb semantics for english-chinese translation," *Translation*, vol. 10, pp. 59–92, Mar. 1995.
[35] H. Panetto and J. Cecil, "Information systems for enterprise integration, interoperability and networking: Theory and applications," *Enterp. Inf. Syst.*, vol. 7, no. 1, pp. 1–6, Feb. 2013.
[36] C. Pautasso, "RESTful Web service composition with BPEL for REST," *Data Knowl. Eng.*, vol. 68, no. 9, pp. 851–866, Sep. 2009.
[37] C. Pautasso, O. Zimmermann, and F. Leymann, "Restful web services vs. 'big' web services: Making the right architectural decision," in *Proc. 17th Int. World Wide Web Conf. 2008,* Beijing, China, pp. 805–814.
[38] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *J. Artif. Intell. Res.*, vol. 11, pp. 95–130, Jul. 1999.
[39] G. G. Richard III, "Service advertisement and discovery: Enabling universal device cooperation," *IEEE Internet Comput.*, vol. 4, no. 5, pp. 18–26, Oct. 2000.
[40] J. Song, S. Han, A. K. Mok, D. Chen, M. Lucas, and M. Nixon, "WirelessHART: Applying wireless technology in real-time industrial process control," in *Proc. RTAS'08,* St. Louis, MO, USA, 2008, pp. 377–386.
[41] W. Wang, S. De, R. Toenjes, E. Reetz, and K. Moessner, "A comprehensive ontology for knowledge representation in the internet of things," in *Proc. TrustCom,* Liverpool, U.K., 2012, pp. 1793–1798.
[42] X. Wang and X. Xu, "DIMP: An interoperable solution for software integration and product data exchange," *Enterp. Inf. Syst.*, vol. 6, no. 3, pp. 291–314, Aug. 2012.
[43] B. M. Wilamowski and O. Kaynak, "Oil well diagnosis by sensing terminal characteristics of the induction motor," *IEEE Trans. Ind. Electron.*, vol. 47, no. 5, pp. 1100–1107, Oct. 2000.
[44] B. M. Wilamowski, R. C. Jaeger, and M. O. Kaynak, "Neuro-fuzzy architecture for CMOS implementation," *IEEE Trans. Ind. Electron.*, vol. 46, no. 6, pp. 1132–1136, Dec. 1999.
[45] J. Wan and J. Jones, "Managing IT service management implementation complexity: From the perspective of the Warfield version of systems science," *Enterp. Inf. Syst.*, vol. 7, no. 4, pp. 490–522, 2013.
[46] L. Xu, H. Liu, K. Wang, and S. Wang, "Modeling and analysis techniques for cross-organizational workflow systems," *Syst. Res. Behav. Sci.*, vol. 26, no. 3, pp. 367–389, 2009.
[47] L. Xu, "Enterprise systems: State-of-the-art and future trends," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 630–640, Nov. 2011.

**Guangyi Xiao** (M'10) received the Master's degree in software engineering from the University of Macau, Macau, China, in 2009. Currently, he is a Ph.D. student in e-commerce technology with the Department of Computer and Information Science, University of Macau.

His research interests include controlled vocabulary and business documents, mainly applied to the fields of e-commerce, e-marketplace, and virtual world.

**Jingzhi Guo** (M'05) received the Ph.D. degree in Internet computing and e-commerce from Griffith University, Brisbane, Australia, in 2005; the M.Sc. degree in computation from the University of Manchester, Manchester, U.K., in 2010; and the B.Econ. degree in international business management from the University of International Business and Economics, Beijing, China, in 1988.

Currently, he is an Associate Professor in e-Commerce Technology with the University of Macau, Macau, China. His research interests include concept representation, semantic integration, and collaboration systems, mainly applied to the fields of e-commerce, e-marketplace, e-banking, and the virtual world.

**Li Da Xu** (M'86–SM'11), photograph and biography not available at the time of publication.

**Zhiguo Gong** (M'10) received the B.S. and M.S. degrees in mathematics from Heibei Normal University, Hebei, China, in 1983, and Peking University, Beijing, China, in 1998, respectively, and the Ph.D. degree in computer science from the Department of Computer Science, Chinese Academy of Science, Beijing, China, in 1998.

Currently, he is an Associate Professor and Head of Computer Science with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include databases, digital library, web information retrieval, and web mining.