Toward Open Facility Networking: Semantics Management for Higher-Level Interoperability

Hideya Ochiai
The University of Tokyo / NICT
7-3-1 Hongo, Bunkyo
Tokyo, 113-8656
+81-3-5841-6748, JP
jo2lxq@hongo.wide.ad.jp

Hiroshi Esaki
The University of Tokyo / NICT
7-3-1 Hongo, Bunkyo
Tokyo, 113-8656
+81-3-5841-6748, JP
hiroshi@wide.ad.jp

ABSTRACT

Open facility networking enables the increase of density, scalability and flexibility at the deployment of sensors and actuators, which are essentially required in Green IT scenarios. We, in this paper, propose ubiquitous directory in open and multi-domain facility networking framework as a semantics management system for those sensors and actuators. Ubiquitous directory achieves interoperability between sensors, actuators and other software components at their semantic level. We developed a prototype implementation in Green UT project, and confirmed that sensors and actuators could be accessed by managed and interoperable rules, even over the operational domain boundaries, which could not have been done before introducing the ubiquitous directory scheme.

Categories and Subject Descriptors

C.2.3 [**Computer Communication Networks**]: Network Operations – *network management*

General Terms

Management, Standardization

Keywords

Facility Networking, Semantics Management, Multi-Domain Sensor Networks

1. INTRODUCTION

Facility networking in buildings, factories and houses is widely acknowledged as a promising technology for energy saving or for reduction of energy wastes. The major changes from the traditional facility networking to the energy-aware facility networking are (1) analytical works on wider range of dataset, (2) density of deployed sensors and actuators, (3) flexibility of network configurations and (4) intelligence of control. (5) Real-time control is less important, in energy-saving scenarios.

These requirements have lead to three-tiered architecture (Figure 1): i.e., (i) field-buses (e.g., BACnet[1], Lonworks[2]) at the bottom-tier, (ii) data storage at the middle-tier and (iii) application programs at the top-tier. Gateways at field-buses submit sequence of data and events to the data storage. Application programs retrieve them from the storage, and make analytical works and produce new field-bus configurations that optimize the control to save energy. In this architecture, we assume that field-bus gateways, data storage, and applications be networked by TCP/IP. Each component are developed by different vendors and operated by different companies.

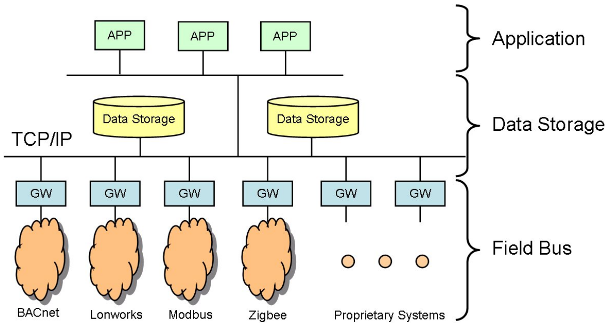


Figure 1. Three-tiered facility networking architecture.

Applications carry out analytical works and configuration optimizations, in this architecture, by retrieving data from gateways and storages and by writing the configuration into gateways. Here, in practice, without understanding the semantic information or the background knowledge of data points (e.g., where the sensor is deployed, what it monitors, how frequently it produces data sequence), applications cannot take any actions in finding appropriate sensors and in choosing appropriate algorithms to make the analyses and the optimizations.

In this paper, we propose ubiquitous directory (UD) that globally manages the semantic information of sensors and actuators. The semantic information we discuss in this paper is a collection of static models of the real world that the applications aware. For instance, it has a model of locations in a building (e.g., an entity in this model may represents *around the entrance of room 102*), a model of units (e.g., *kWh consumed in the past one day*) and a model of target objecta (e.g., *microwave oven*, or *65-inch plasma display*). Ubiquitous directory manages those semantic entities, and system operators bind data points (i.e., sensors and actuators) to these semantic entities so that applications and any other system components could identify the semantics of the data points.

Some works (e.g., GSN[3], Live E![4], WWSN[5]) have proposed and discussed frameworks for sharing sensor data over multiple operational domains. However, to our knowledge, sharing semantic information is not deeply discussed, which is absolutely necessary to enable enough interoperability among different organizations.

This paper describes a prototype experience on UD construction as a case study made in Green UT project[6] which has 1609 data points.

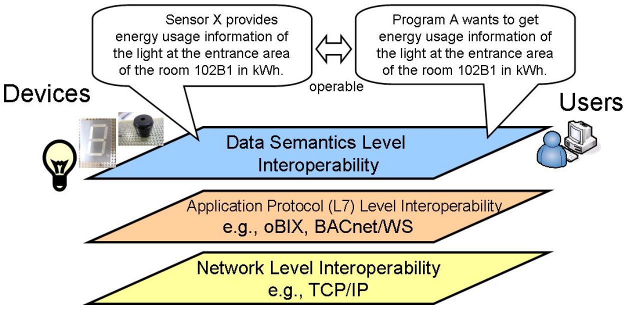


Figure 2. Layers of interoperability.

This paper is organized as follows. In section 2, we describe interoperability issues in facility networking. In section 3, we propose the semantics management scheme by introducing ubiquitous directory. Section 4 describes implementation and working experiences. In section 5, we make discussion. Finally, section 6 provides the conclusion of this paper.

2. LAYERS OF INTEROPERABILITY

There are many types of interoperability that should be maintained to enable multi-domain sensor networking. In this work, as figure 2 illustrates, we divide the interoperability level into three levels: (1) network level interoperability, (2) application protocol level interoperability and (3) data semantics level interoperability. Network level interoperability can be easily achieved, for example, by networking devices by a TCP/IP network. Application protocol level interoperability can be made by using the same application layer protocols. The existing facility networking application layer protocols are open building information exchange (oBIX)[7], BACnet/WS[1] and Live E![5]. The most important interoperability level, which we focus on in this paper, is data semantics level. The background knowledge or properties of data points (e.g., where the sensor is deployed, what it monitors, how frequently it produces data sequence) must be shared among multiple operational domains.

As Sandra has described [8], semantic interoperability must be based on semantic agreements among multiple domains with regard to algorithms for computing requested values, the expected side effects of a requested procedure, or the source or accuracy of data elements. Most of the semantic information remains to be implicit to other domains, which makes semantic agreements among them difficult. He mentioned that semantic agreements require the involvement of people for those reasons and that describing them explicitly as a metadata might help detecting mismatch of semantics but still difficult to completely maintain consistency of semantics among multiple domains.

We basically agree to this idea, and we try to develop a framework that enables or helps semantic interoperability in the area of facility networking. In our approach, we develop a semantic world that application programs aware in a so called ubiquitous directory (UD). Operators associate data points with the managed semantic world, which will improve semantics consistency level among multiple domains. Translating data from one semantic world to another semantic world, for example, temperature degree from Celsius to Fahrenheit, could be one of the goals of such a framework. However, we do not deeply discuss this case in this paper, because we believe that it can be made possible after we have developed a semantic information management framework that we focus on in this paper.

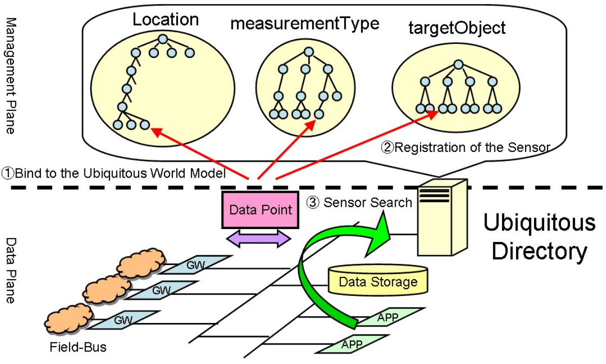


Figure 3. Semantics management by ubiquitous directory.

3. SEMANTICS MANAGEMENT BY UBIQUITOUS DIRECTORY

3.1 Architecture

Figure 3 shows the architecture which explicitly manages the semantics of data points. We added ubiquitous directory to the three-tiered architecture that we have presented in figure 1. The ubiquitous directory provides a management service that enables interoperability in data semantics level.

The ubiquitous directory has multiple application domains that represent the real world; e.g., in figure 3, location model, measurementType model and targetObject model are managed. A data point physically attached at a field-bus is logically handled as an element of data and works in the data-plane (i.e., among field-buses, data storages and applications). By binding the data point to these models, applications can find the point by appropriate queries. Here, the binding algorithm can be implemented as follows;

(1) Field-bus operators set their data points to have pointers (i.e., links) to the entities of these modeled worlds.

(2) The field-bus gateways send a request to register their data points to the entities.

(3) The entities know what points belong to them.

In this way, data points and the managed entities in UDs can be linked to each other.

3.2 Application Domain Modeling

At first, the semantic world should be modeled and shared over operational domain boundaries in order to maintain interoperability at the data semantics level. We propose a method of modeling the semantic world by application domains, which model should be managed in UDs.

Figure 4 presents the example of location, measurementType and targetObject application domain models managed in a UD. The location domain has the physical location entities that applications refer in order to identify where data points are deployed. The measurementType domain manages the data type classes that applications must aware in data processing. The targetObject domain manages the classes of facilities that applications must

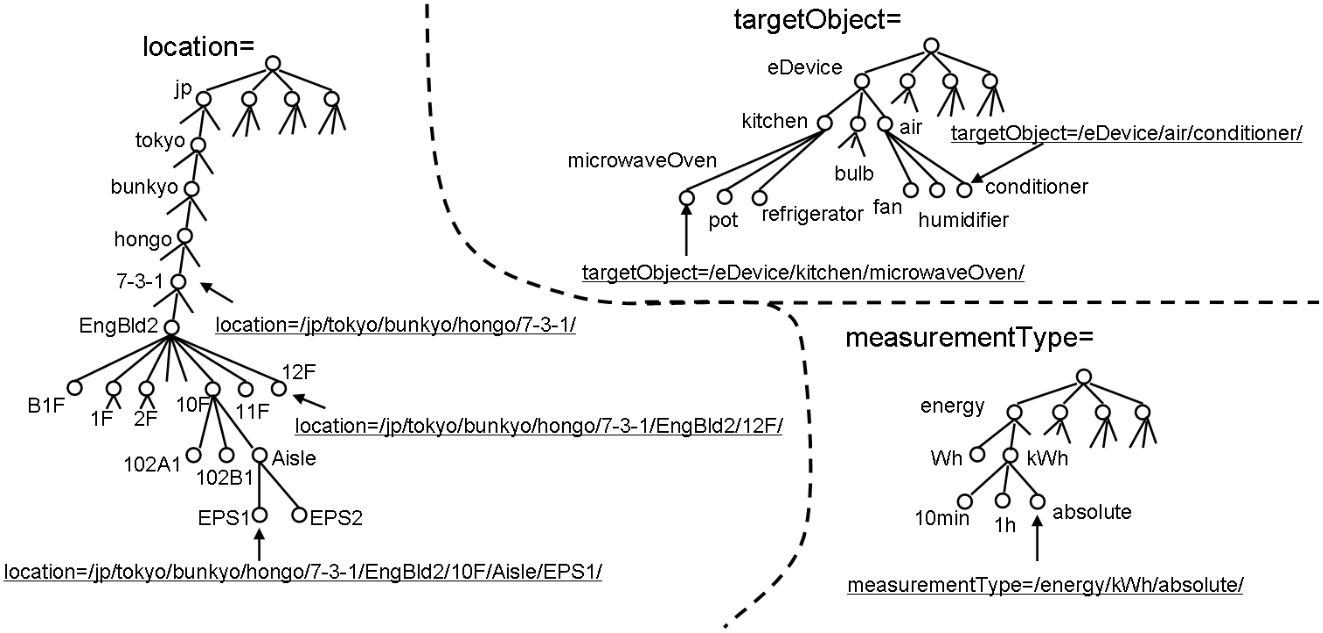


Figure 4. Application domain examples for location, targetObject and measurementType.

understand to present users what it monitors. Every application domain is modeled on a tree-based data structure.

An entity has a global unique name to identify the semantics from everywhere in the system. For example, when we describe,

```
targetObject="/eDevice/kitchen/microwaveOven/"
```

it means a type of the facilities, which exactly is microwave oven categorized to the class of kitchen in the electronic devices. Another example is,

```
measurementType="/energy/kWh/absolute/"
```

This means a type or unit of sensor readings that a data point observes, which exactly is consumed energy in kWh from when the data point has been setup.

In this paper, we have presented the three types of application domains as an example. However, depending on what the users want to do with the facility networking system, there can be other types of application domains, for example, operator's domain, pointVendor's domain and employee's domain. UDs are designed to manage every domain independently, which enables the extension or inclusion of unexpected application semantic world.

It is certainly true that, as each application domain is modeled independently, the data structure can be also modeled independently; e.g., (1) in expressing the name of city and street, we might use tree-based data structure such as location="/jp/tokyo/bunkyo/hongo/7-3-1/"; (2) in expressing the geographical location point we might use two-dimensional data structure with latitude and longitude coordinate, for example, geoLocation="(39.28312, 135.534)". However, we believe that simplicity and extensibility are quite important regarding to the feasibility of system deployment and operation. From our previous working experiences, managing semantics by different data structure does not work. Thus, application domain designers should model the world only by tree-based data structure in practice.

3.3 Binding Semantics to a Data Point

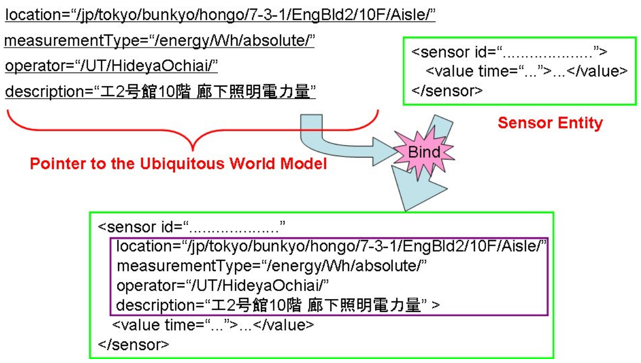
Next, data points should be bound to the managed models to identify the semantics of them from anywhere in the system.

Figure 5 illustrates the binding process of semantics to a data point. A data point is described as an XML entity, and the semantics are attached by XML attributes. In this way, a data point entity can have pointers to the managed models.

A data point has its unique identifier so that it can be identified and looked up at the data-plane. As for the identification of semantics, applications check the attached attributes.

This binding process should be done at the gateways when it comes out from the field-bus. Based on the pointers presented in the XML, the UD registers data points to the managed application domain entities, and maintains the consistency of semantics description over multiple operational domains.

If the described pointer is not managed in the UD, the pointer is misconfigured or the UD is not sufficiently configured. The misconfiguration of pointer happens, for example, when system operator puts with typo: e.g., microwareOven for microwaveOven. In this case, the semantics in the operator's mind certainly managed in the UD, and the operator can fix the description. Through, this verification process, the system gets description consistency for the same semantics entity, and increases the semantic level interoperability.



```
<sensor id="...................">
   <value time="...">...</value>
</sensor>
```

```
<sensor id="..................."
   location="/jp/tokyo/bunkyo/hongo/7-3-1/EngBld2/10F/Aisle/"
   measurementType="/energy/Wh/absolute/"
   operator="/UT/HideyaOchiai/"
   description="工2号館10階 廊下照明電力量" >
  <value time="...">...</value>
</sensor>
```

Figure 5. Binding semantics to a data point.

A UD sometimes need to be re-configured or extended in more detail. For example, a field-bus operator puts a sensor in room 203, but the entity that represents room 203 is not registered in the UD. In such cases, the UD is not sufficiently configured, and it must be extended so that room 203 should be managed before binding the pointer to a data point.

4. IMPLEMENTATION AND EXPERIENCES

We have made a prototype facility networking system in Green UT project. We have setup a Live E! data server that collects data from seven field-bus domains (operated by seven organizations). We have installed a viewer as an application that (1) lists up data points in Web browsers and that (2) shows sequence of data by graph. The total number of data points in the system is 1609.

We designed and implemented three application domains in an Excel-based UD, and we (operators) bound the data points to those managed application domains. We did not implement data point registration method in UD because our main concern is the management and operational costs. Besides, application could lookup data points by semantic-based queries with our Live E! system once the semantic pointers were attached to the data points

In this section, we describe (1) the issues we experienced before carrying out any semantics management, (2) the effect of semantics management by an UD, and (3) the cost and the feasibility of the management.

4.1 Without Semantics Management

Without managing semantics (i.e., before introducing an UD), the following three types of issues were observed.

(1) Operators have described each data point differently in their own manner, which made applications difficult to search the data points.

(2) Different field-buses used different expressions for data values.

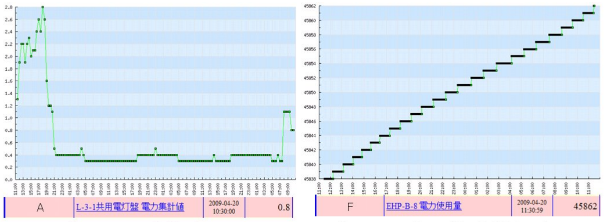
(3) Even for the same category's data point, the detailed meanings were different among different companies

Administrator	Point
A	102B1室内機-1 電力集計値
B	10F江崎研実験室102B1－冷蔵庫
D	102B1江崎研究室電流
F	10F江崎教授室 温度(1)
C	学生室照明①電力量

Figure 6. Description of the same room by different operators.

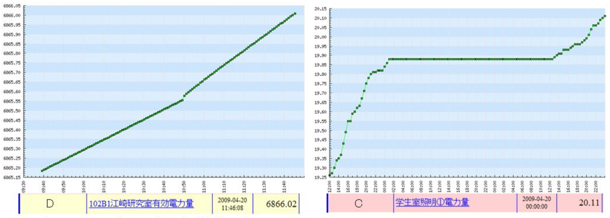
Administrator	Point	Time	Value
D	91A4客員教授室ロスナイ状態	2009-04-20 10:48:09	true
D	91B1会議室ロスナイ状態	2009-04-20 10:48:09	false
E	3階 231講義室前　廊下-照明	2009-04-20 10:48:01	T
E	3階 機会系会議室前　廊下-照明	2009-04-20 10:48:01	F
F	EHP-B-8 運転状態	2009-04-20 10:47:59	運転

Figure 7. Description of system status by different operators.



Summary of 10min;
Measure by 0.1kWh

Summary from the initial setup;
Measure by 1kWh



Summary from the initial setup;
Measure by 0.01kWh

Summary from the initial setup;
Measure by 0.01kWh

Figure 8. Data sequence of consumed energy.

Figure 6 illustrates that operators described data points differently even though all of them are deployed in the same room (102B1). This heterogeneity brings chaos – applications cannot easily find the points by a simple ruled query. This is an example of a lack of interoperability among operational domains at the semantic level.

Figure 7 shows that different field-buses report the working status by their own presentation rules. Operator D used *true* and *false* in expressing the status, operator E used *T* and *F*, and operator F used *working* and *stopped* in Japanese letter.

As in figure 8, to present the consumed electrical energy, some data points made the summary from when the facility had setup, however, others (e.g., top-left side in figure 8) made the summary from only the last 10 minutes. The granularity was also different among different system operational domains.

```
<sensor comment="102B1室内機-1 電力量集計値"
      id="live-e://esp2.elab.ic.i.u-tokyo.ac.jp/00015/00021/01486/PPE0“
      location="/jp/tokyo/bunkyo/hongo/7-3-1/EngBld2/10F/102B1/“
      measurementType="/energy/kWh/10min/“
      operator="/cimx/">
    <value time="2009-07-13T06:30:00.0000000+09:00">0</value>
</sensor>
…
<sensor comment="10F江崎研実験室102B1ー冷蔵庫“
     id="live-e://p100vmst.gutp.ic.i.u-tokyo.ac.jp:8080/watt/services/BACnet…
     location="/jp/tokyo/bunkyo/hongo/7-3-1/EngBld2/10F/102B1/"
     measurementType="/energy/Wh/absolute/"
     operator="/panasonic/sakaguchi.k/">
    <value time="2009-07-13T07:32:45.1375410+09:00">433281</value>
</sensor>
…
<sensor comment="102B1江崎研究室電流“
     id="live-e://lmj-obix-server.gutp.ic.i.u-tokyo.ac.jp/obix/ut_hongo/t2/102…
     location="/jp/tokyo/bunkyo/hongo/7-3-1/EngBld2/10F/102B1/“
     measurementType="/current/A/"
     operator="/yokogawa/Hiroaki.Tanaka/" >
    <value time="2009-07-13T07:33:01.6488140+09:00">3.61</value>
</sensor>
…
<sensor comment="10F江崎研究室 温度(1)"
     id="live-e://azbil-bacnetws.gutp.ic.i.u-tokyo.ac.jp/WebServices/BACnet…
     location="/jp/tokyo/bunkyo/hongo/7-3-1/EngBld2/10F/102B1/“
     measurementType="/temperature/C/"
     operator="/yamatake/fujimura-fumio/">
    <value time="2009-07-13T07:33:01.9743560+09:00">26</value>
</sensor>
…
<sensor comment="学生室照明①電力量"
     id="live-e://pew-iiu-integral.gutp.ic.i.u-tokyo.ac.jp/Saver1/PEW0005"
     location="/jp/tokyo/bunkyo/hongo/7-3-1/EngBld2/10F/102B1/"
     measurementType="/energy/kWh/absolute/"
     operator="/panasonic-denko/takazoe/" >
    <value time="2009-07-13T00:00:00.0000000+09:00">65.5</value>
</sensor>
```

Figure 9. Retrieved data points with observed data

4.2 The Effect of Semantics Management by an Ubiquitous Directory

After introducing an UD, and with some operational effort (i.e., modeling of application domains and binding semantics to data points), the first issue that description difference of data points among field-bus operators was solved.

Applications could find data points by semantic-based queries. For example, we have got 45 data points by the query below:

```
location=
  "/jp/tokyo/bunkyo/hongo/7-3-1/EngBld2/10F/102B1/"
```

Figure 9 presents the examples of those data points. This result shows that users could find data points over operational domain boundaries by specifying the location in the managed and shared location model.

In this way, the first issue was solved. However, the second and the third issues were not solved only from this approach. There should be other rules that define (1) the manner of expression of data values and (2) output sequence scheme. To homogenize the data expression manners and other differences, we must develop standardized rules in the system and operators must reconfigure each field-bus to obey the standard to enable semantics interoperability.

Table 1. Time spent for ubiquitous directory configuration

Application Domain	Scale [size]	Config [min]
location	114	120
operator	7	10
measurementType	133	155

Table 2. Time spent for binding semantics to data points

Field-bus	Points	Locations	Types	Config[min]
A	679	9	8	60
B	5	1	1	4
C	14	2	1	10
D	349	23	7	65
E	40	33	1	17
F	37	8	5	18
G	382	42	13	110

4.3 Management and Binding Cost

In managing semantics of data points by UDs, we must (1) design and implement application domains in UDs and (2) bind data points to the entities in UDs. In this study, we evaluated the cost of maintenance with regard to the time spent in the configuration.

Table 1 describes the time spent in configuring the UD: i.e., the configuration time of setting application domain models. We have implemented three application domains. The *Scale* column presents the number of elements that application domain has. The *Config* column presents the time spent in designing and setting the application domain.

Table 2 describes the time of binding *location* and *measurementType* to data points. We had seven field-buses named A to G here. The *Points* column presents the number of data points in the field-bus. *Locations* and *Types* columns describe the number of related application domain entities to the field-bus; location domain and measurementType domain respectively. The *Config* column shows the time spent in binding pointers to those data points. For example, at field-bus=D, 349 data points were operated, and they were classified into 23 entities in the location application domain and 7 entities in the measurementType application domain. 65 minutes was spent in binding those data points to the application domains.

5. DISCUSSION

In designing the application domains and binding data points to them, we referred to the information submitted by each field-bus operator. This implies that configuration of UD is request-based by field-bus operators. Field-bus operators with assuming some application scenarios must at first request the configuration of their UD if appropriate entities are not managed in it. After the UD administrator has registered the requested entities, the field-bus operators can bind their data points to the newly managed entities at the UD.

Applications we have tested in this paper are just a viewer that lists up all the data points and draws graphs. When we would assume other kind of applications, it would be quite certain that we must reconfigure the UD to also match the applications' requirements. Our final target applications are energy saving related applications. Thus, we must develop them and test the feasibility and effect of UD-based semantics management.

6. CONCLUSION

We have studied semantics management in multi-domain facility networking, which is necessary to enable interoperability of sensors, actuators and other software components at the semantic level. By sharing semantic information (e.g., what sensors are managed in which domain, or what actions actuators do), the interoperability should be guaranteed among different domains.

We have introduced *ubiquitous directory (UD)* into the three-tiered facility networking architecture. We also introduced the concept of application domains to model the semantic world of applications in UDs. By binding data points to the managed and shared semantic world, semantic level interoperability should be increased.

We have implemented a prototype system in the Green UT project, and tested the impact of the proposed schemes. Before introducing an UD, every operator has described their data points, defined the expression rule of data values and the properties of sequential data in their own manner. After introducing an UD, we experienced that applications could resolve the data points by managed query expressions.

In the current status, we have not considered many application scenarios in constructing the UD. Since the application domains in UD are related to the applications on the framework, we should test with more application scenarios to evaluate whether our proposed approach is practically feasible or not.

7. REFERENCES

[1] BACnet, http://www.bacnet.org/

[2] Lonworks, http://www.echelon.com/

[3] Green University of Tokyo Project, http://www.gutp.jp/

[4] Kerl Aberer, Manfred Hauswirth and Ali Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks", In Proceedings of IEEE MDM, 2007.

[5] Hideya Ochiai, Satoshi Matsuura, Hideki Sunahara, Masaya Nakayama and Hiroshi Esaki, "Operating architecture and multi-attribute search for wide area sensor networks", IEICE Transaction on Communications, Vol.J91-B, No.10, pp.1160--1170, October, 2008

[6] Lei Shu. Manfred Hauswirth, Long Cheng, Jian Ma, Vinney Reynolds and Lin Zhang, "Sharing worldwide sensor networks", In Processdings of IEEE SAINTW, 2008.

[7] Open Building Information Xchange(oBIX), http://www.obix.org/

[8] Sandra Heiler, "Semantic Interoperability", ACM Computing Surveys, Vol. 27, No. 2, pp.271--173, Jun. 1995.