Journal of Universal Computer Science, vol. 16, no. 13 (2010), 1745-1760
submitted: 25/5/10, accepted: 6/5/10, appeared: 1/7/10 © *J.UCS*

On Sustainability of Context-Aware Services Among Heterogeneous Smart Spaces

Jason J. Jung
(Knowledge Engineering Laboratory
Department of Computer Engineering
Yeungnam University, Korea
j2jung@{intelligent.pe.kr, gmail.com})

Abstract: Most of ambient intelligence studies have tried to employ inductive methods (e.g., data mining) to discover useful information and patterns from data streams on sensor networks. However, since the spaces have been sharing their information with each other, it is difficult for such inductive methods to conduct the discovery process from the sensor streams intermixed from the heterogeneous sensor networks. In this paper, we propose an ontology-based middleware system to improve sustainability of context-aware service in the interconnected smart spaces. Two main challenges of this work are *i*) sensor data preprocessing (i.e., session identification) and *ii*) information fusion (i.e., information integration). The ontology in each sensor space can provide and describe semantics of data measured by each sensor. By aligning these ontologies from the sensor spaces, the semantics of sensor data captured inside can be compared. Thus, we can find out not only relationships between sensor streams but also temporal dynamics of a data stream. To evaluate the proposed method, we have collected sensor streams from in our building during 30 days. By using two well-known data mining methods (i.e., co-occurrence pattern and sequential pattern), the results from raw sensor streams and ones from sensor streams with preprocessing were compared with respect to two measurements recall and precision.
Key Words: Ontology; Semantic sensor networks; Preprocessing; Stream mining.
Category: H.1.1, H.3.5, I.2.11

1 Introduction

A variety of types of sensors (and sensing technologies) have been developed and implemented into sensor networks, which are capable of capturing environmental data for monitoring a certain space. For example, a wireless sensor network has been installed in a number of environments including a forest [Culler et al. 2004]. Consequently, people can expect to figure out the relationships between the status of ecosystem and climate changes by analyzing many environmental data (e.g., humidity, temperature, and so on). It means that as more diverse sensors are developed, the chance on understanding and analyzing the environments might be getting much higher.

In terms of service provision to users within the sensor spaces, people have been looking for more contextually relevant services via their own mobile devices [Chen et al. 2004, Ejigu et al. 2007]. Thereby, ontologies have been exploited to represent various contexts from users and spaces [Euzenat et al. 2008, Gu et al. 2004, Narayanan et al. 2007, Roussaki et al. 2006, Wagner et al. 2004]. One of the main purposes of such (wireless or ubiquitous) sensor networks is to keep track of activities and behaviors of people inside the space. We may thus expect pervasive services, i.e., people can be "anytime and anywhere" provided with more relevant information and services by installing more sensors in the sensor network. There have been several practical projects to develop and investigate ontology-based sensor networks, e.g., semantic sensor web [Sheth et al. 2008] and SensorWare[1].

Furthermore, similar to [Jabeur et al. 2009], the sensor spaces can be integrated to exchange the information they have collected, as shown in Fig. 1. As a result, each sensor space can be expected to use the information collected in other sensor networks, so that this sensor network can efficiently provide the pervasive services to the users who are firstly visiting there.

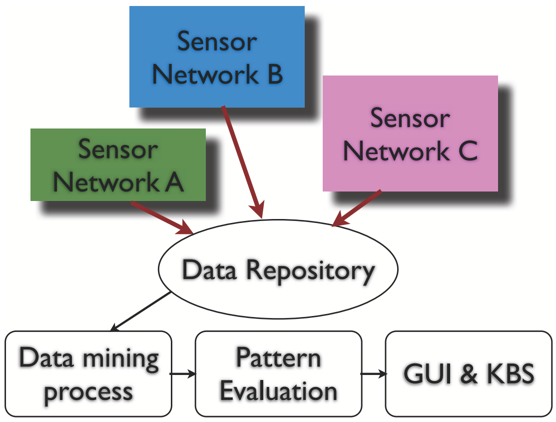

Figure 1: Conventional architecture for mining sensor streams collected from multiple sensor networks

Thereby, inductive methods (e.g., data mining) have to be exploited to discover useful patterns from the integrated sensor streams. However, there are some difficulties on such discovery process. The problems that we are focusing on in this paper are as follows:

– Noisy and redundant data from sensors

– Semantic heterogeneity between sensor spaces

[1] SensorWare Systems. http://www.sensorwaresystems.com/

Of cause, these problems might be caused by many unexpected reasons (e.g., sensor malfunction). In this work, we want to deal with these problems by preprocessing the raw data streams from the sensor networks, and also expect to improve the performance of data mining methods for extracting useful knowledge which happens in the environments.

In this paper, we propose an ontology-based *preprocessing* method and information fusion method for implementing an information integration platform with heterogeneous sensor streams. The ontology is employed to bridge multiple sensor networks which are semantically heterogeneous with each other. Two main goals of this work are

- session identification of sensor streams annotated by the ontology, and
- information fusion for integrating rules discovered from the sensor spaces.

The outline of this paper is as follows. In Sect. 2, we show how conventional sensor spaces can exploit the ontologies for providing pervasive services. Sect. 3 addresses semantic annotation scheme for sensor streams, and Sect. 4explains semantic identification method based on similarity measurement between sensor streams. To evaluate the performance of the proposed approach, Sect. 5 illustrates the experimental results and discusses the important issues that we have realized from the experiments, and compare the proposed approach to the existing ones. Finally, Sect. 6 draws a conclusion of this study.

2 Ontology-based Sensor Space

Basically, the sensor nodes are installed into a certain environment, and the information collected from the sensor nodes are used to understand and detect significant patterns in the environment. However, in many cases (particularly, tracking people), the information from a single sensor network is not enough for this process. Ontology-based sensor network has been investigated to solve this problem by integrating several sensor networks [Jabeur et al. 2009]. As shown in Fig. 2, the ontologies in sensor spaces are aligned with each other, in advance, for realizing semantic heterogeneity between sensor data. Thus, the system can automatically justify whether a service is contextually relevant to certain users or not.

2.1 Context Ontology for Pervasive Intelligence

The main purpose of the ontologies is to annotate sensor streams, i.e., describe what semantics are related to information collected from the corresponding sensor. Context ontology we are using in this work contains environmental concepts and the relationship between the concepts.

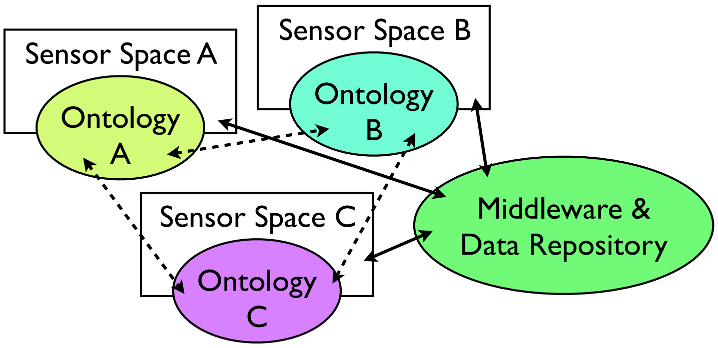

Figure 2: Ontology-based sensor space

Definition 1 (Context ontology). Context ontology $\mathbb{O}$ is represented as

$$\mathbb{O} := (\mathcal{C}, \mathcal{R}, \mathcal{E}_{\mathcal{R}}, \mathcal{I}_{\mathcal{C}}) \quad (1)$$

where $\mathcal{C}$ and $\mathcal{R}$ are a set of classes (or concepts), a set of relations (e.g., equivalence, subsumption, disjunction, etc), respectively. $\mathcal{E}_{\mathcal{R}} \subseteq \mathcal{C} \times \mathcal{C}$ is a set of relationships between classes, represented as a set of triples $\{\langle c_i, r, c_j\rangle | c_i, c_j \in \mathcal{C}, r \in \mathcal{R}\}$, and $\mathcal{I}_{\mathcal{C}}$ is a power set of instance sets of a class $c_i \in \mathcal{C}$.

Table 1: An example of contextual ontologies

```
<owl:Class rdf:ID="Environment" />
<owl:Class rdf:ID="EnvironmentalCondition" />
<owl:Class rdf:ID="Humidity">
    <rdfs:subClassOf rdf:resource="#EnvironmentalCondition" />
</owl:Class>
<owl:Class rdf:ID="Noise">
    <rdfs:subClassOf rdf:resource="#EnvironmentalCondition" />
</owl:Class>
<owl:Class rdf:ID="Temperature">
    <rdfs:subClassOf rdf:resource="#EnvironmentalCondition" />
</owl:Class>
<owl:Class rdf:ID="Lighting">
    <rdfs:subClassOf rdf:resource="#EnvironmentalCondition" />
</owl:Class>
<owl:ObjectProperty rdf:ID="hasEnvironmentalCondition">
    <rdfs:domain rdf:resource="#Environment" />
    <rdfs:range rdf:resource="#EnvironmentalCondition" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="MeasuringTemporature">
    <rdfs:domain rdf:resource="#Temporature" />
<rdfs:range rdf:resource="#Thermometer" />
</owl:ObjectProperty>
```

While there are various kinds of ontology development methods, we have implemented the context ontology by extending the existing ontologies (e.g., SUMO[2] and SOUPA [Chen et al. 2004]) based on TOVE ontology methodology [Fox 1992]. We have chosen 24 scenarios and organized them in a hierarchical form. Similar to [Eid et al. 2007], some upper-level concepts are derived from SUMO. Thus, as shown in Table 1, the context ontology, called ContextOnto[3], is written in OWL[4]. Especially, this ontology includes a number of physical sensor nodes by `Owl:Individual`. Table 1 shows only a part of ContextOnto.

2.2 Ontology alignment

In order to solve the heterogeneity drawbacks, discovery process for significant alignments between ontologies needs to be automated. A set of given sensor space ontologies have to be matched as finding out the best configuration of alignments between ontology entities (e.g., classes and properties). We assume that the best configuration should be maximizing the summation of class similarities. Similarity between two classes is computed by not only class labels but also neighbor classes (i.e., subclasses and superclasses) and the annotation instances related to the class. To preprocess a set of annotation instances, some of terms in the annotation instances should be extracted and regarded as principal components representing the class [Jung 2008c].

2.2.1 Alignment based on class similarity

In order to find optimal alignment between two ontologies, we have to measure the similarity between classes consisting of the ontologies.

Definition 2 (Class similarity). Given a pair of classes from two different ontologies, the class similarity ($Sim_{\mathcal{C}}$) between c and c' is defined as

$$Sim_{\mathcal{C}}(c,c') = \sum_{E \in \mathcal{N}(C)} \pi_E^C MSim_Y(E(c), E(c')) \tag{2}$$

where $\mathcal{N}(C) \subseteq \{E^1 \ldots E^n\}$ is the set of all relationships in which the classes participate (for instance, subclass, superclass, or instances). We have to consider on three components $Y = \{L, C, I\}$ $i)$ class labels (L), $ii)$ neighboring classes (C), and $iii)$ annotation instances (I). The weights π_E^C are normalized (i.e., $\sum_{E\in\mathcal{N}(C)} \pi_E^C = 1$). Class similarity measure $Sim_{\mathcal{C}}$ is assigned in $[0, 1]$.

As a matter of fact, a similarity function between two set of classes can be established by finding a maximal matching maximizing the summed similarity between the classes:

$$MSim_C(S,S') = \frac{\max(\sum_{\langle c,c'\rangle \in Pairing(S,S')} (Sim_{\mathcal{C}}(c,c'))}{\max(|S|,|S'|)}, \tag{3}$$

[2] Suggested Upper Merged Ontology (SUMO). http://www.ontologyportal.org/
[3] ContextOnto. http://intelligent.pe.kr/SemSensorWeb/ContextOnto.owl
[4] Web Ontology Language (OWL). http://www.w3.org/TR/owl-features

in which $Pairing$ provides a matching of the two set of classes. Methods like the Hungarian method allow to find directly the pairing which maximizes similarity. The algorithm is an iterative algorithm that compute this similarity [Euzenat and Valtchev 2004]. This measure is normalized because if $Sim_{\mathcal{C}}$ is normalized, the divisor is always greater or equal to the dividend.

In this work, we have to take in account all possible relationships (r and $r^{\star}$) between classes for $\mathcal{N}(C) = \{E^{sup}, E^{sub}, E^{equ}\}$, provided $i)$ the superclass (E^{sup}) and the subclass (E^{sub}) defined in each ontology fragment, and $ii)$ the equivalent class (E^{equ}) by manual alignments of human experts, respectively. Then, Eq. 2 can be rewritten as:

$$\begin{aligned} Sim_{\mathcal{C}}(c,c') = \pi_L^C sim_L(L(c), L(c')) \\ + \pi_{sub}^C MSim_{\mathcal{C}}(E^{sub}(c), E^{sub}(c')) + \pi_{sup}^C MSim_{\mathcal{C}}(E^{sup}(c), E^{sup}(c')) \\ + \pi_{equ}^C MSim_{\mathcal{C}}(E^{equ}(c), E^{equ}(c')) + \pi_{\mathcal{I}}^C Sim_{\mathcal{I}}(I(c), I(c')) \end{aligned} \tag{4}$$

where the set functions $MSim_{\mathcal{C}}$ compute the similarity of two entity collections. Label similarity sim_L is simply computed by string matching algorithms such as *Levenshtein* edit distance [Levenshtein 1996], substring distance [Euzenat 2004], and so on. Similarity measure between two classes can be turned into a distance measure $Distance = 1 - Similarity$ by taking its complement to 1.

Especially, in order to enhance the accuracy of the class similarity, the last term in Eq. 4 is representing instance-level similarity measurement between business process annotations. We exploit three different heuristic functions, and they are formulated by

$$Sim_{\mathcal{I}}(I(c), I(c')) = \frac{N}{\max(|I(c)|, |I(c')|)} \tag{5}$$

$$= \max_{n=1}^{N} Sim_{\langle \mathcal{P}_\alpha, \mathcal{P}_\beta \rangle \in Pairing(I(c), I(c'))}(L(\mathcal{P}_\alpha), L(\mathcal{P}_\beta))_n \tag{6}$$

$$= \frac{\sum_{n=1}^{N} Sim_{\langle \mathcal{P}_\alpha, \mathcal{P}_\beta \rangle \in Pairing(I(c), I(c'))}(L(\mathcal{P}_\alpha), L(\mathcal{P}_\beta))_n}{N} \tag{7}$$

where N is the number of pairs of term features whose distances computed by string matching methods are less than threshold τ_{Dist}, i.e.,

$$EditDistance(L(\mathcal{P}_\alpha), L(\mathcal{P}_\beta))_{\mathcal{P}_\alpha \in I(c), \mathcal{P}_\beta \in I(c')} \leq \tau_{Dist}. \tag{8}$$

Three equations are denoted as H_1, H_2, and H_3, and they return the normalized number of matched pairs of terms, the maximum similarity among matched terms, and the average similarity of matched terms, respectively. Because instance-level class similarity can uncover the latent semantic information of the classes, the normalization process with the weighting factor is expected to prune incorrect alignments between them.

As a result, the proposed alignment process between heterogeneous ontologies can be represented as a set of pairs of classes from two different ontologies. We refer a class pair to *correspondence* (e.g., equivalence or subsumption).

Definition 3 (Alignment). Given two sensor space ontologies $\mathcal{O}_i$ and $\mathcal{O}_j$, the alignments between both ontoliges are represented as a set of correspondences $CRSP_{ij} = \{\langle c, r, c' \rangle | c \in \mathcal{O}_i, c' \in \mathcal{O}_j\}$ where r means the relationship between c and c', by maximizing the summation of class similarities $\sum Sim_{\mathcal{C}(c,c')}$.

2.3 Sensor Stream Mining

By discovering a certain sequential pattern from a data stream, we can realize and conduct trend and periodicity analysis over time [Han and Kamber 2006]. Particularly, this method has been applied to web browsing sequences for eliciting personal interests [Jung 2005a]. In this study, we focus on applying sequential pattern mining [Agrawal and Srikant 1995, Srikant and Agrawal 1996] to discover *frequent* sequential patterns from the sensor streams. For example, in Table 2, five kinds of sensor nodes have been sampled from timestamp T_0 to T_8.

Table 2: An example of sensor streams

Timestamp	RFID Reader	Temperature (°C)	Lights (On/Off)	Air Conditioner (Switch)	Project Screen (Pull-down)
T_0	U_1	27	Off	Off	False
T_1	U_1	28	On	On	False
T_2	U_1, U_2	30	Off	On	True
T_3	U_1, U_2, U_3	28	Off	Off	True
T_4	U_1, U_2, U_3	27	On	On	False
T_5	U_2, U_3	25	On	On	False
T_6	U_2	25	On	On	False
T_7	U_2	26	On	Off	False
T_8	U_1, U_2	26	Off	Off	True

To do the discovery process and more importantly improve the performance of this process, the sensor streams should be preprocessed by removing noises and missing data. The preprocess is done by session identification (also called sessionization) for segmenting the sequences of streams with respect to contextual situations.

3 Semantic Annotation for Sensor Streams

Data streams collected from sensor nodes have to be annotated by using context ontologies (e.g., ContextOnto) This semantic annotation is to derive relevant semantic metadata from the ontologies for making the semantic streams understandable.

In other words, the sequences of sensor data are transformed into the sequences of RDF metadata. Through this transformation by annotation, the relationships (i.e., semantic similarity) between RDF metadata can be measured. Especially, the heterogeneity of integration between sensor networks can be easily and efficiently dealt with. As an example in Table 3, the semantic annotation are represented by RDF. It illustrate that environmental context "Temperature = 28.0" at "Timestamp = T_1."

Table 3: An example of semantic annotation of a sensor stream

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://intelligent.pe.kr/SemSensorWeb/2007/03/ContextOnto.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:units="http://intelligent.pe.kr/SemSensorWeb/2007/03/Units.owl#"
  xml:base="http://intelligent.pe.kr/SemSensorWeb/2007/03/ContextOnto.owl">
<Temperature rdf:ID="Temp8">
  <Cxt_Property_7 rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
    28.0
  </Cxt_Property_7>
</Temperature>
<Time rdf:ID="T1">
  <hasTemperature rdf:resource="#Temp8"/>
</Time>
</rdf:RDF>
```

Another benefit obtained from semantic annotation is integration between heterogeneous sensor networks shown in Fig. 2. By referring to a set of correspondences $CRSP_{ij} = \{\langle c, r, c' \rangle | c \in \mathcal{O}_i, c' \in \mathcal{O}_j\}$ by ontology alignment, the sensor streams from multiple sensor networks are integrated as shown in Fig. 3. These sensor networks must be connected to a middleware for transmitting the data to be aggregated into the stream repository [Heinzelman et al. 2004]. According to the number of installed sensors and sampling frequencies, a centralized sensor stream repository receives and store unique amount of information from each sensor network over time. Furthermore, any external information sources on the web can be interoperated to the middleware for enriching the integrated sensor streams.

The integration between sensor streams can be achieved by merging RDF annotations. This work assumes that only one ontologies are exploited to all sensor

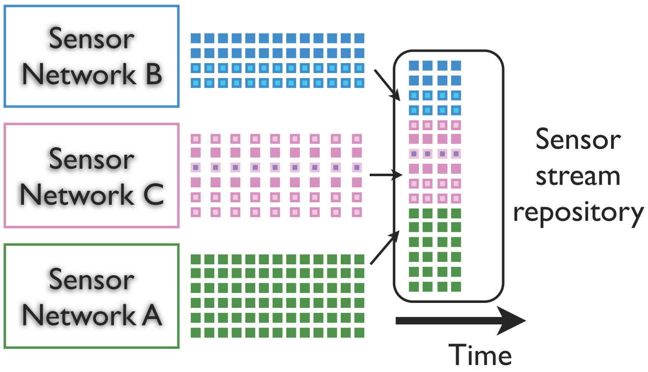

Figure 3: Integration of semantic sensor streams

networks, while there can be distributed ontologies (i.e., each sensor network has its own ontology). If there are several different ontologies, we have to consider automated ontology mapping methods to find semantic correspondences between the ontologies [Jung 2008a].

4 Session Identification as Preprocessing

Even though we have enough information by integrating sensor streams from multiple environments, it is impossible to directly apply conventional data mining algorithms to the integration sensor streams. Thereby, we are focus on conduct session identification [Jung 2005b], with respect to contextual relationships between timestamped sensor data.

Generally, there have existed two heuristic-based approaches for the session identification. as follows;

– time-oriented heuristic, which sessionizes with a predefined time interval, and

– frequency-oriented heuristic, which sessionizes with a frequency of sensor streams (i.e., the time interval dynamically changes).

Differently from them, we propose a novel ontology-based session identification method [Jung 2008b]. The main idea of this method is to compute and observe the statistical distribution of semantic similarity for finding out semantic outliers, as shifting a window W along to the sensor stream. Thereby, firstly, given two timestamps t_i and t_j, we have to compute the semantic similarity $\Delta^{\diamond}$ between two RDF annotations $\mathbb{A}(t_i)$ and $\mathbb{A}(t_j)$. Jung [Jung 2007] has explained several methods to measure the semantic similarities between two RDF metadata. Semantic similarity matrix $D_{\Delta^\diamond}$ can be given by

$$D_{\Delta^\diamond}(t,t') = \begin{pmatrix} \dots & \dots & \dots \\ \dots & \Delta^\diamond[\mathbb{A}(t_i), \mathbb{A}(t_j)] & \dots \\ \dots & \dots & \dots \end{pmatrix} \quad (9)$$

where the size of this matrix is the predefined time interval $|W|$ and the diagonal elements are all zero. For example, given a sensor stream in Table 2, we can obtain the semantic similarity matrix as shown in Table 4.

Table 4: An example of semantic similarity matrix (Size of the sliding window is 3.)

	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
T_0	0	0.8	0.3						
T_1	0.8	0	0.25	0.15					
T_2	0.3	0.25	0	0.78	0.26				
T_3		0.15	0.78	0	0.32	0.36			
T_4			0.26	0.32	0	0.87	0.92		
T_5				0.36	0.87	0	0.89	0.88	
T_6					0.92	0.89	0	0.95	0.24
T_7						0.88	0.95	0	0.23
T_8							0.24	0.23	0

Based on a semantic similarity matrix $D_{\Delta^\diamond}$, the semantic mean $\mu^\diamond$ is given by

$$\mu^\diamond(t_1, \dots, t_T) = \frac{2\sum_{i=1}^{T}\sum_{j=i}^{T} D_{\Delta^\diamond}(i,j)}{T(T-1)} \quad (10)$$

where $D_{\Delta^\diamond}(i,j)$ is the (i,j)-th element of semantic similarity matrix. This is the mean value of upper triangular elements except diagonals. Then, with respect to the given time interval T, the semantic deviation $\sigma^\diamond$ is derived as shown by

$$\sigma^\diamond(t_1, \dots, t_T) = \sqrt{\frac{2\sum_{i=1}^{T}\sum_{j=i}^{T} \left(D_{\Delta^\diamond}(i,j) - \mu^\diamond(t_1, \dots, t_T)\right)^2}{T(T-1)}} \quad (11)$$

These factors are exploited to quantify the semantic similarity between two random RDF annotations of sensor stream and statistically discriminate semantic outliers such as the most distinct or the N distinct data from the rest in the range of over preset threshold, with respect to given time interval. For instance, the semantic similarity matrix in Table 4 can be represented as Fig. 4 by observing the $\mu^\diamond$ and $\sigma^\diamond$ over time.

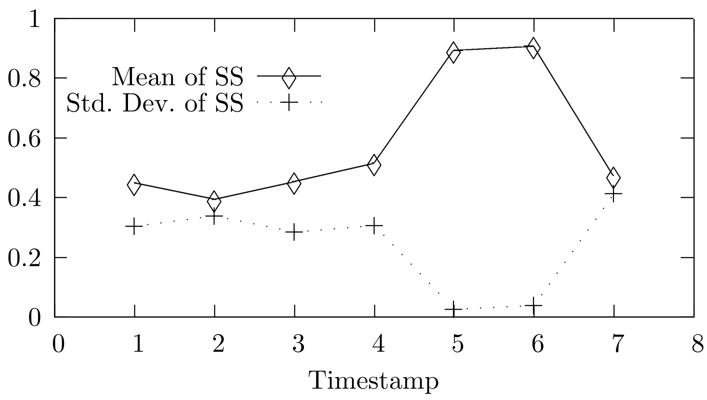

Figure 4: Distribution of semantic similarities

Based on the observed dynamics of the distribution (i.e., Fig. 4), we have to investigate how the sensor streams are segmented (i.e., session identification). Four heuristics have been considered that a new session begins in the RDF annotation stream.

- H_{1-1}. when the mean of semantic similarity $\mu^\diamond$ is less than a threshold τ_μ
- H_{1-2}. when the standard deviation of semantic similarity $\sigma^\diamond$ is more than a threshold τ_σ,
- H_{2-1}. when $\mu^\diamond$ turns from upward to downward,
- H_{2-2}. when $\sigma^\diamond$ turns from downward to upward,

In consequence, from Fig. 4, these four heuristics can detect the specific moment when (or where) should be divided from the sequence. Table 5 depicts the results on session identification by the four heuristics. We will empirically evaluate these heuristics, and discuss which heuristics are better than others later.

Table 5: Results on session identification by the four heuristics

Heuristics	H_{1-1}	H_{1-2}	H_{2-1}	H_{2-2}
Session identification between	(T_1, T_2)	(T_6, T_7)	(T_6, T_7) (T_6, T_7)	(T_1, T_2), (T_3, T_4) (T_6, T_7)

Finally, we can discover frequent sequential patterns by applying data mining algorithms to the sessionized data sets. It was difficult to do this from raw sensor streams in Table 2. Once we have the sessionized result (i.e., Table 5), we can easily find out the frequent patterns and also be aware of behavioral patterns of people in the particular environment. Thus, in this case, we have discovered and translated the following pattern (with support and confidence), as follows;

- Project Screen Pull-Down (True) → Light (Off) [20%, 95%]
- When project screen gets pull-down in this room, light switch will be turned off.

5 Experimental results and discussion

In order to evaluate the proposed preprocessing method, we have installed four sensor networks, which are composed of five kinds of sensor nodes, into three lecture room and one corridor. The sensor streams have been collected for about one month (from 9 March 2009 to 6 April 2009).

As a first experiments, we had to set up the length of sliding windows. Also, we needed to investigate whether (and how) the length of sliding windows affects to the performance of sessionization. Thus, as changing the length of sliding window, we have measured the numbers of sessions identified and the numbers of patterns discovered, and then justified the validity of the patterns. The results are shown in Table 6.

Table 6: Evaluation of performance of session identification with respect to length of sliding window

Length of sliding window (minute)	Number of sessions	Number of discovered patterns	Performance of validity
1	73,192,929,039	15	67.7%
2	6,853,742,836	14	68.5%
3	1,958,289,740	11	70.5%
5	368,526,283	7	71.4%
10	40,102,492	4	86.3%
30	2,510,275	3	90.5%
60	231,022	3	94.0%

As the length of window is longer, the numbers of sessions and patterns are exponentially decreasing, but the human validity is gradually increasing. We found out that it results from long-term patterns, which means more general patterns. It makes human evaluators to determine positive validity on the patterns.

Second issue is to evaluate four heuristics (i.e., H_{1-1} to H_{2-2}) which have been designed for the session identification. We used 30 minute-size sliding window for this. As shown in Table 7, heuristic H_{2-1} outperforms the others, with respect to the validity testing. More importantly, we found out that the distributions of means of semantic similarity (i.e., H_{1-1} and H_{2-1}) is more crucial than those of standard deviation of semantic similarity (i.e., H_{1-2} and H_{2-2}), respectively.

Table 7: Evaluation of performance with respect to the heuristics

	H_{1-1}	H_{1-2}	H_{2-1}	H_{2-2}
Number of sessions	2510275	2058426	1360231	1074583
number of patters	4	5	4	4
Validity	90.45%	84.7%	**93.3%**	81.5%

Third evaluation issue is to compare the proposed preprocessing method with other methods. Once we have had the frequent sequential patterns discovered by [Agrawal and Srikant 1995], we have compared the proposed method (i.e., M_{Onto}) with the existing sessionization ones (i.e., M_{Generic}, M_{Time}, and M_{Sampling}) with respect to the number of the discovered patterns and the precision of the patterns. The precision (i.e., "agree" and "disagree" scores) has been indicated by human experts on this experiments.

- M_{Generic} has no sessionization step. It assumes that the contexts are always identical.
- M_{Time} uses a time slot to sessionize. The size of the time slot is fixed.
- M_{Sampling} is to conduct a random sampling. It is expected to efficiently work on a large-scale streams.

The result is shown in Table 8. M_{Onto} proposed in this paper has outperformed the other three methods by discovering more patterns. More significantly, in terms of "agree" score, the patterns discovered by M_{Onto} have shown higher precision than pattern by M_{Generic} (by 295.6%), M_{Time} (by 119%), M_{Sampling} (by 142.8%). But, interestingly, the disagree score of M_{Onto} has shown slightly higher than that of M_{Sampling}. We think that M_{Sampling} has been able to discover the most stable and plain patterns from the sensor streams.

Table 8: Comparison of performances of stream preprocessing methods (The size of sliding window for M_{Onto} is 30-minute, and we chose H_{2-1} for session identification.)

		$M_{Generic}$	M_{Time}	$M_{Sampling}$	M_{Onto}
Number of patterns		4	4	4	7
Precision	Agree	1 (25%)	3 (60%)	2 (50%)	5 (71.4%)
	Disagree	2 (50%)	2 (40%)	1 (25%)	2 (28.6%)
	Do not know	1 (25%)	0 (0%)	1 (25%)	0 (0%)

6 Concluding Remark and Future Work

As a conclusion, we have claimed that heterogeneous sensor streams from multiple environments should be efficiently integrated and preprocessed for better data mining performance. It was difficult to do this from raw sensor streams in Table 2. Once we have the sessionized result (i.e., Table 5), we can easily find out the frequent patterns and also understand the contextual sequences of people in the particular environment. Especially, this study will give good chance to other works, because there have been many interesting practical projects to develop and investigate ontology-based sensor networks, e.g., semantic sensor web [Sheth et al. 2008] and SensorWare.

In future work, for better sessionization performance, combination between several heuristics (e.g., H_{2-1} and H_{2-2}) will be conducted, because these heuristics can make up for weak points with each other.

We will focus on developing a real sensor network application by applying several sequential pattern mining approaches to discover the sequential patterns. Moreover, by matching distributed ontologies, most of sensor network systems on open network will be integrated [Jung 2010]. The ontologies will be play an important role of logical reasoner in this area.

Acknowledgement

This work was supported by the Korean Science and Engineering Foundations (KOSEF) grant funded by the Korean government (MEST). (No. 2009-0066751).

References

[Agrawal and Srikant 1995] Agrawal, R. and Srikant, R.: Mining sequential patterns. In Philip S. Yu and Arbee L. P. Chen, editors, *Proceedings of the 8th International Conference on Data Engineering, March 6-10*, pages 3–14. IEEE Computer Society, 1995.

[Chen et al. 2004] Chen, H., Perich, F., Finin, T., and Joshi, A.: Soupa: Standard ontology for ubiquitous and pervasive applications. In *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04)*, pages 258–267. IEEE Computer Society, 2004.

[Culler et al. 2004] Culler, D. E., Estrin, D., and Srivastava, M. B.: Overview of sensor networks. *Computer*, 37(8):41–49, 2004.

[Eid et al. 2007] Eid, M., Liscano, R., and El Saddik, A.: A universal ontology for sensor networks data. In *Proceedings of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications Ostuni- Italy, 27-29 June*, 2007.

[Ejigu et al. 2007] Ejigu, D., Scuturici, M., and Brunie, L.: An ontology-based approach to context modeling and reasoning in pervasive computing. In *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*, pages 14–19. IEEE Computer Society, 2007.

[Euzenat 2004] Euzenat, J.: An API for ontology alignment. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *Proceedings of the 3rd International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 698–712. Springer, 2004.

[Euzenat et al. 2008] Euzenat, J., Pierson, J., and Ramparany, F.: Dynamic context management for pervasive applications. *The Knowledge Engineering Review*, 23(1):21–49, 2008.

[Euzenat and Valtchev 2004] Euzenat, J., and Valtchev, P.: Similarity-based ontology alignment in OWL-Lite. In Ramon López de Mántaras and Lorenza Saitta, editors, *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'2004), Valencia, Spain, August 22-27, 2004*, pages 333–337. IOS Press, 2004.

[Fox 1992] Fox, M. S.: The tove project towards a common-sense model of the enterprise. In Fevzi Belli and Franz Josef Radermacher, editors, *Proceedings of the 5th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-92), June 9-12*, volume 604 of *Lecture Notes in Computer Science*, pages 25–34, Paderborn, Germany, 1992. Springer.

[Gu et al. 2004] Gu, T., Wang, X. H., Pung, H. K., and Zhang, D. Q.: An ontology-based context model in intelligent environments. In *Proceedings of International Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2004.

[Han and Kamber 2006] Han, J. and Kamber, M.: *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.

[Heinzelman et al. 2004] Heinzelman, W. B., Murphy, A. L., Carvalho, H. S., and Perillo, M. A.: Middleware to support sensor network applications. *IEEE Network*, 18(1):6–14, 2004.

[Jabeur et al. 2009] Jabeur, N., McCarthy, J. D., Xing, X., and Graniero, P. A.: A knowledge-oriented meta-framework for integrating sensor network infrastructures. *Computers & Geosciences*, 35(4):809–819, 2009.

[Jung 2005a] Jung, J. J.: Collaborative web browsing based on semantic extraction of user interests with bookmarks. *Journal of Universal Computer Science*, 11(2):213–228, 2005.

[Jung 2005b] Jung, J. J.: Semantic preprocessing of web request streams for web usage mining. *Journal of Universal Computer Science*, 11(8):1383–1396, 2005.

[Jung 2007] Jung, J. J.: Exploiting semantic annotation to supporting user browsing on the web. *Knowledge-Based Systems*, 20(4):373–381, 2007.

[Jung 2008a] Jung, J. J.: Ontology-based context synchronization for ad-hoc social collaborations. *Knowledge-Based Systems*, 21(7):573–580, 2008.

[Jung 2008b] Jung, J. J.: Query transformation based on semantic centrality in semantic social network. *Journal of Universal Computer Science*, 14(7):1031–1047, 2008.

[Jung 2008c] Jung, J. J.: Taxonomy alignment for interoperability between heterogeneous virtual organizations. *Expert Systems with Applications*, 34(4):2721–2731, 2008.

[Jung 2010] Jung, J. J.: Reusing ontology mappings for query segmentation and routing in semantic peer-to-peer environment. *Information Sciences*, 180(17):3248–3257, 2010.

[Levenshtein 1996] Levenshtein, I.V.: Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8):707–710, 1996.

[Narayanan et al. 2007] Narayanan, S., Sievers, K., and Maiorano, S. J.: Occam: Ontology-based computational contextual analysis and modeling. In Boicho N. Kokinov, Daniel C. Richardson, Thomas Roth-Berghofer, and Laure Vieu, editors, *Proceedings of the 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2007), Roskilde, Denmark, August 20-24*, volume 4635 of *Lecture Notes in Computer Science*, pages 356–368. Springer, 2007.

[Roussaki et al. 2006] Roussaki, I., Strimpakou, M., Pils, C., Kalatzis, N., and Anagnostou, M.: Hybrid context modeling: A location-based scheme using ontologies. In *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06).* IEEE Computer Society, 2006.

[Sheth et al. 2008] Sheth, A., Henson, C., and Sahoo, S. S.: Semantic sensor web. *IEEE Internet Computing*, 12(4):78–83, 2008.

[Srikant and Agrawal 1996] Srikant, R., and Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In Peter M. G. Apers, Mokrane Bouzeghoub, and Georges Gardarin, editors, *Proceedings of the 5th International Conference on Extending Database Technology (EDBT'96), March 25-29*, volume 1057 of *Lecture Notes in Computer Science*, pages 3–17, Avignon, France, 1996. Springer.

[Wagner et al. 2004] Wagner, M., Noppens, O., Liebig, T., Luther, M., and Paolucci, M.: Semantic-based service discovery on mobile devices. In *Proceedings of the Demo Track of the 4th International Semantic Web Conference (ISWS 2004)*, 2004.