International Journal of Informatics Society

08/12 Vol. 4 No. 2 ISSN 1883-4566

Informatics Society

Aims and Scope

The purpose of this journal is to provide an open forum to publish high quality research papers in the areas of informatics and related fields to promote the exchange of research ideas, experiences and results.

Informatics is the systematic study of Information and the application of research methods to study Information systems and services. It deals primarily with human aspects of information, such as its quality and value as a resource. Informatics also referred to as Information science, studies the structure, algorithms, behavior, and interactions of natural and artificial systems that store, process, access and communicate information. It also develops its own conceptual and theoretical foundations and utilizes foundations developed in other fields. The advent of computers, its ubiquity and ease to use has led to the study of informatics that has computational, cognitive and social aspects, including study of the social impact of information technologies.

The characteristic of informatics' context is amalgamation of technologies. For creating an informatics product, it is necessary to integrate many technologies, such as mathematics, linguistics, engineering and other emerging new fields.

Guest Editor's Message

Jun Sawamoto
Guest Editor of the Eleventh Issue of International Journal of Informatics Society

We are delighted to have the eleventh and special of the International Journal of Informatics Society (IJIS) published. This issue includes selected papers from the Fifth International Workshop on Informatics (IWIN2011), which was held in Venezia, Italy, Sep 16 - 21, 2011. The workshop was held at Ca' Foscari University. This workshop was the fifth event for the Informatics Society, and was intended to bring together researchers and practitioners to share and exchange their experiences, discuss challenges and present original ideas in all aspects of informatics and computer networks. In the workshop, 27 papers were presented at seven technical sessions. The workshop was complete in success. It highlighted the latest research results in the area of networking, business systems, education systems, design methodology, groupware and social systems.

Each IWIN2011 paper was reviewed in terms of technical content and scientific rigor, novelty, originality and quality of presentation by at least two reviewers. From those reviews, 14 papers are selected for publication candidates of IJIS Journal. This tenth includes four papers of them. The selected papers have been reviewed from their original IWIN papers and accepted as publication of IJIS. The papers were improved based on reviewers' comments.

We hope that the issue would be of interest to many researchers as well as engineers and practitioners in this area.

We publish the journal in print as well as in an electronic form over the Internet. This way, the paper will be available on a global basis.

Jun Sawamoto is currently a Professor of Faculty of Software and Information Science, Iwate Prefectural University, Japan. He received the B.E. and M.E. in mechanical engineering from Kyoto University in 1973 and 1975. He joined Mitsubishi Electric Corporation in 1975. He received his MSCE degree from Stanford University in 1986. He received his PhD from Tokyo Denki University in 2004. He was a senior researcher at the Institute for New Generation Computer Technology of Japan in 1985–1988. His research interests include ubiquitous computing, human-interface system, multi-agent systems, and cooperative problem solving. He is a member of IPSJ, IEICE, IEEJ, IEEE-CS, ACM.

A Study on Stream Prediction Based on Timing Correlation among Multiple Data Stream

Kenji Arai[†], Yoh Shiraishi[‡], and Osamu Takahashi[‡]

[†]Graduate School of Systems Information Science, Future University Hakodate, Japan
[‡]School of Systems Information Science, Future University Hakodate, Japan
{g2110002[†], siraisi[‡], osamu[‡]}@fun.ac.jp

Abstract - Research on data streams has attracted a great deal of attention in many fields in recent years, such as sensor-network technologies and stock quote data. Therefore, stream prediction technologies have attracted the attention of stream mining technologies. When we want to obtain the predicted value of a certain single data stream, most methods use past data on the data stream. However, we think that correlations, such as synchronization, can be used for a method for predicting streams, and their accuracy might be better than methods that only use past data on single data streams. In addition, we need to take into consideration that correlations are not based on synchronization, which we call "similarity correlation". We suggest a method for detecting "timing correlation" from multiple data streams in this paper, and a method for predicting streams on the basis of these correlations. We also demonstrate and discuss the efficacy of these methods.

Keywords: Data Stream, Clustering, Classification, Correlation, Prediction

1 INTRODUCTION

Research on data streams has attracted a great deal of attention in numerous fields in recent years, such as sensor-network technologies and stock quote data. Data streams are expressed as time-series data of unlimited length and increase in real time. Stream mining technologies have been studied intensively [1] to find significant patterns from data streams. For example, some researchers have studied the diverse trends in data streams [2]. These trends express many features, such as periodicity. They are used to predict future data of data streams in accordance with past and present trends.

When predicting certain data stream, most methods only use past data on the predicted object. However, if two or more data streams are measured from sensors installed indoors, their correlation, such as synchronization of values, might appear from these data streams. We think their correlations can be used by methods for predicting streams and they might be more accurate than approaches that only use past data on single data stream. We assumed a situation where two data streams were measured simultaneously in Fig. 1 as an example of this theory. The wave forms of temperature data change in accordance with humidity data near the current time. The accuracy of prediction may decrease when strange values like these are measured if we only use a method for prediction using past data on single data streams. However, if the correlation in which temperature data increase after a rapid increment in humidity is found, accuracy of prediction will be improved by taking into consideration this correlation.

We assumed we needed to consider two correlations between data streams.

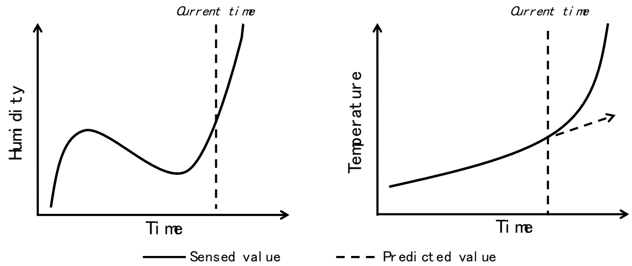

Figure 1: Sample of Stream Prediction Based on Correlation

I. Similarity Correlation

Streams A and B in Figure 2 are measured simultaneously. The two data streams are similar because of their measured values and their trends are very close. There is a correlation based on similarity between these two data streams which we call the "similarity correlation" in this paper.

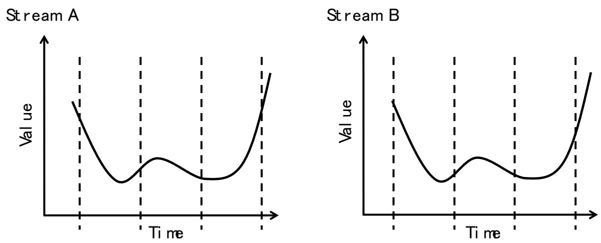

Figure 2: Example of Similarity Correlation

II. Timing Correlation

Streams C and D in Fig. 3 are measured simultaneously.

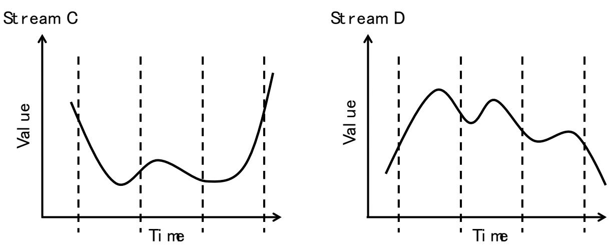

Figure 3: Example of Timing Correlation

The measured value and trend of stream C differ from those of stream D. However, we assumed there was a rule based on the timing of changing trends. For example, when stream C increases, stream D decreases. We found this rule to be one of correlation and call it "timing correlation" in this paper.

Our task was to predict data streams on the basis of similarity and timing correlation, and we archived this by using our new approach.

2 RELATED WORK

2.1 Related Work in Stream Mining

Many researchers in stream mining technologies have studied trend detection technologies, such as detection trends from partial sequences made from data streams [3][4]. Trends represent features, such as the periodicity of partial sequences and data streams. Kawashima et al. found a method for reducing the number of calculations by cutting off the dimensions of partial sequences with adaptive piecewise constant approximation [5]. They succeeded in fast matching of partial sequences to sample sequences in the database with this method. Toyota et al. introduced a method for detecting trends based on the dynamic time warping (DTW) distance [6]. Papadimitriou et al. [7] predicted future data of data streams by using models like the auto regression model using the wavelet coefficients of data streams. The coefficients of wavelet transforms and model updates are executed each time a stream data is detected at minimum cost because previous results are used. However, this method does not consider the correlation between multiple data streams.

In addition, Sakurai et al. introduced a correlation detection technology called "BRAID" [8]. Zhu created StatStream [9] to detect correlations by comparing the DFT coefficients of partial sequences. However, these methods consider similarity correlation, timing correlation. Moreover, they did not refer to stream prediction.

2.2 Relationship with Data Mining

Data mining technologies are similar to those for stream mining. They find a significant rule from time-series data stored at databases. Stream mining technologies are regarded as one field in data mining technologies. However, stream mining technologies must consider the amount of memory and processing time, because data streams are time-series data that increase at certain intervals in real time. Therefore, some methods in stream mining technologies are different from these in data mining technologies. In stream mining technologies, the processing time and amount of memory are more important than complete results. In addition, approximated solutions are generically used instead of exact solutions in real time processes. Therefore, incremental algorithms are valued because they decrease the number of calculations. These algorithms use previous results to calculate new results.

3 PROPOSED METHOD

3.1 Overview of Proposed Method

We explain the requirements for our approach to solve our tasks in this section.

Data streams are predicted from estimation of future trends by taking into account past trends in data streams. Consequently, a method is required for detecting trends and correlations from past measured data. In addition, the method for detecting correlations needs to be based on numerous features of data streams to detect timing correlations, and it cannot use degrees of similarity such as the DFT coefficient and DTW distance. Therefore, these methods need to be incremental and only use a certain amount of memory.

First, we will explain the environment for data streams in our approach. Two or more data streams are measured simultaneously. All data in the data streams are measured accurately at certain intervals with no missing or delayed data.

Our method for detecting correlations and predicting data streams is clarified in what follows. Our method detects trends from data stream in real time. Next, it manages information on the appearance of trends. Correlation in data streams is detected by matching detected trends and the number of relationship on the basis of past trends. After that, our method predicts future data streams in accordance with correlations between data streams. It does the following:

1. Detects new trends.
2. Manages information on the appearance of trends.
3. Detects correlations between data streams, and
4. Predicts future data stream based on correlations.

3.2 Detection of Trends

A) System for Classifying Trends

Our method divides data streams into partial sequences in real time, and detects trends in partial sequences as current trends in the data streams by using a classification system. Therefore, a classification system must be prepared from past data. The method executes the following five steps to construct a classification system.

i. Divides all data streams into partial sequences by a certain interval.
ii. Extracts feature quantity patterns FQ-P and FQ-N from all partial sequences.
iii. Cluster partial sequences in accordance with FQ-P.
iv. Cluster partial sequences in associated with each cluster in accordance with FQ-N.
v. Construct a classifier from the clusters.

a) Divide Streams into partial sequences

Our method divides each data stream into multiple partial sequences with a certain interval. Each interval is called a window. We have assumed partial sequences divided by the same window have the same number of data. All data streams are divided into the same number of partial sequences.

b) Extract Feature Quantity Patterns

The method extracts feature quantities that express trends in all partial sequences and bundles a number of feature quantities as a pattern called a feature quantity pattern in this paper.

We used the power spectrum obtained from the DFT coefficients of partial sequences and natural values of partial sequences as resources for feature quantities. We called a feature quantity pattern obtained from the former an "FQ-P". We also called the latter an "FQ-N". The method extracts the following.

- Maximum Value / Data index of maximum value
- Minimum Value / Data index of minimum value
- Variance
- Value of integral

FQ-P expresses the periodicity of partial sequences while FQ-N expresses other waviness features.

c) Clustering Partial Sequences

Our method classifies partial sequences into various groups (clusters) in accordance with the extracted feature quantities (feature quantity pattern). A wide variety of feature quantity patterns can be generated from partial sequences. Therefore, detecting the number of clusters in advance is difficult. We must use clustering algorithm that can automatically detect the number of clusters. In addition, the clustering algorithm must consider similarity of classified partial sequences in a cluster, such as those in Fig. 4.

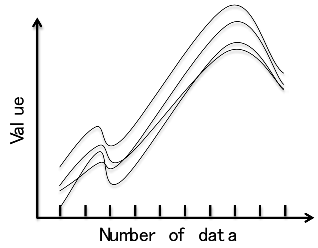

Figure 4: Example of Partial Sequences in Cluster

Therefore, we devised a clustering algorithm that uses a cluster division standard based on Eqs. (1) and (2). In these equations, variable *k* is the number of cluster, variable *n* is the number of partial sequences in cluster *k*, and variable *m* is the number of feature quantities in a feature quantity pattern. Variable x is the variable *j* th value in the *i* th partial sequences in cluster *k*. Variable V_{kj} is the variance of the *j* th values obtained from all partial sequences in cluster *k*. Variance *E* is the standard deviation of V_{kj}, and this is the cluster division standard. The variable *E* of a cluster expresses the variability of partial sequences.

$$V_{kj} = \sum_{i=1}^{n} x_{kij}^2 / n - \left(\sum_{i=1}^{n} x_{kij} / n \right)^2 \quad (1)$$

$$E_k = \sqrt{\sum_{j=1}^{m} V_{kj}^2 / m - \left(\sum_{j=1}^{m} V_{kj} / m \right)^2} \quad (2)$$

Furthermore, the method uses two feature quantity patterns to classify strictly partial sequences into clusters in accordance with similarity of multi-aspects. First, it uses FQ-P for classification. Second, it classifies partial sequences in each cluster into new clusters in accordance with FQ-N. The clustering algorithm using cluster division standard *E* involves six steps.

i. Classify partial sequences into two clusters by two-means clustering in accordance with FQ-P, and add these clusters to the cluster list.
ii. Classify each cluster into two clusters by two-means clustering. The divided clusters are the "parent cluster", and the new clusters are the "child clusters".
iii. Calculate each cluster's division standard.
iv. If the average of a child cluster's cluster division standard is lower than that of the parent cluster, add child clusters to the cluster list and remove the parent cluster.
v. Repeat steps 2 to 4 while the clusters are divided.
vi. Repeat steps 2 to 5 in accordance with FQ-N

d) Construction of classification system

The method constructs a classification system, to classify partial sequences in real time. A classification system is constructed for all data streams. We used C4.5, which is an algorithm for constructing decision trees implemented in Weka [10] as J48. All feature quantity patterns in each cluster are used for the construction. A decision tree classifies a partial sequence into a cluster in accordance with the thresholds of feature quantities, as shown in Fig. 5, which is created from the feature quantity patterns of stream A.

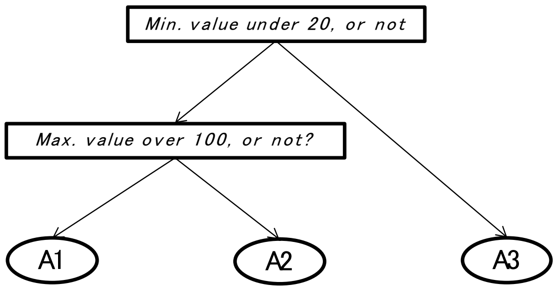

Figure 5: Example of Decision Tree for Feature Quantities

B) Detect Trends using Classification system

Our method detects trends from new partial sequences measured in real time. Each partial sequence obtains a cluster number. These numbers enable partial sequences to be expressed as cluster number streams, as shown in Fig. 6, where the partial sequences of data stream A are classified into {A1, A2, A3}.

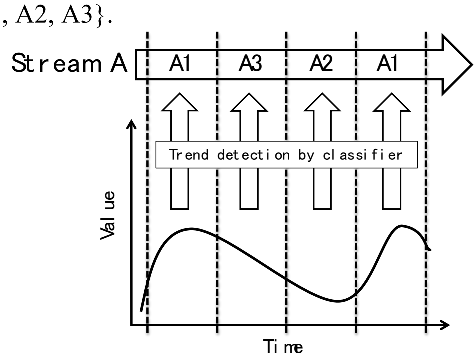

Figure 6: Example of Trend Detection in Stream A

Because the first and fourth partial sequences are similar, the system for stream A classifies both partial sequences into the same cluster, such as A1 in Fig. 6. Other partial sequences are classified into each cluster.

3.3 Managing Correlations

Our method detects the correlation between information on the appearance of trends. Our method manages two kinds of correlation rule between trends. The first is for the correlations between trends from different kinds of data streams. The second is for the correlations between trends in continuing windows from each data stream. We introduced a cluster correlation table to manage the former correlations and a cluster transition table to manage the latter correlations.

A) Cluster Correlation Table

The method detects trends at certain intervals, and a cluster simultaneously appears in all data streams. The cluster correlation table, in Fig. 7, manages the relationships between clusters from different data streams that appear. Stream A and B are measured simultaneously in this figure. Partial sequences from streams A, and B are classified into {A1, A2, A3} and {B1, B2, B3}. For example, when a partial sequence in stream A and partial sequences in stream B are simultaneously classified into A1 for the first and B3 for the second, the corresponding cells (A1, B3) are updated. Each cell means the frequency with which clusters from different data streams appear simultaneously.

		Stream A			Stream B		
		A1	A2	A3	B1	B2	B3
Stream A	A1				5	2	1
	A2				20	20	15
	A3				9	8	38
Stream B	B1	11	20	9			
	B2	29	20	8			
	B3	15	15	38			

Figure 7: Sample of Cluster Correlation Table

Transition rule		Count
From	To	
A1	A1	12
A1	A2	1
A1	A3	33
A2	A1	19
A2	A2	11
A2	A3	4
A3	A1	32
A3	A2	4
A3	A3	5

Figure 8: Example of Cluster Transition Table

B) Cluster Transition Table

We use a cluster transition table to estimate a cluster for a partial sequence in the next window of a stream. A cluster transition table manages the cluster transition rules for each stream. Figure 8 shows an example of a cluster transition table for stream A. When the cluster of the targeted window is A1 and the cluster of the next window is A2, the transition rule is A1→A2. The number of transitions in the corresponding record is updated every time a transition occurs. By referring to this cluster transition table, we can examine a cluster that is easy to change from the targeted cluster. For example, since there are many transitions of A1 →A3 and A3 →A1 in Fig. 8, these transitions are likely to occur.

3.4 Prediction Based on Correlation

A) Prediction Cluster based on Correlation

The method predicts a trend of the next partial sequence on the basis of correlations between clusters. It involves the four steps in Fig. 9.

i. First, it determines the latest window in accordance with the current time. The current time exceeds the end time for the latest window. The current window, including the current time, does not yet have enough data to detect trends with a classifier made for the target stream.
ii. Next, it searches a cluster correlation table for a correlated cluster that is likely to occur simultaneously with the latest cluster from the targeted stream.
iii. It searches the cluster transition table of the stream including the correlated cluster for a cluster (estimated cluster) that will occur in the next window.
iv. Finally, it again searches the correlation table for the cluster of the targeted stream that is likely to occur simultaneously with the estimated cluster. The cluster that is searched is the result of this prediction process.

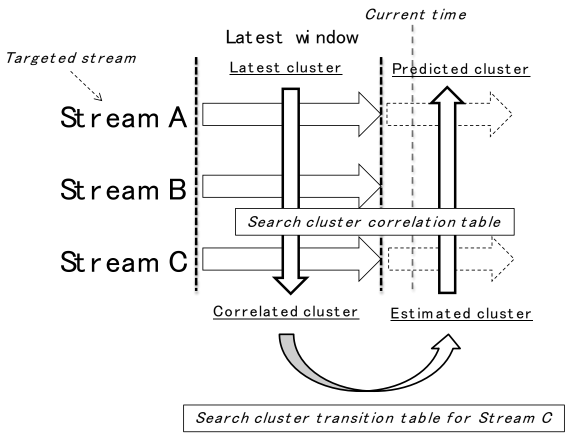

Figure 9: Stream Prediction by Correlation of Clusters

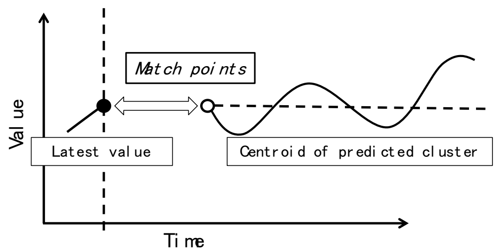

Figure 10: Restoration of sequences by Cluster Centroid

Table 1: Number of Clusters and Evaluation

Sensor	Scale	20 h		10 h		4 h		1 h	
		Num. of Clusters	Evaluation Value	Num. of Clusters	Evaluation Value	Num. of Clusters	Evaluation Value	Num. of Clusters	Evaluation Value
Temperature	0~40C	16	5.212	29	3.494	73	2.680	74	1.570
Humidity	0~100%	20	7.315	28	6.067	85	4.282	264	2.120
Illuminance	0~65000lx	20	1.819	43	1.945	**19**	**11.605**	**5**	**13.173**

Table 2: Average Number of Sequences in Each Cluster

Sensor	20 h	10 h	4 h	1 h
Temperature	3.062	3.370	3.342	12.568
Humidity	2.450	3.500	2.870	3.528
Illuminance	2.450	2.270	**12.842**	**196.000**

B) Partial Restoration of Sequences from Cluster

The method restores the next partial sequences from the predicted cluster in accordance with the latest data and the cluster centroid of the predicted cluster. A cluster centroid is a specific partial sequence created from the average of all partial sequences in a cluster. This means the effective features of partial sequences in the cluster. Our method attaches the latest data to the cluster centroid, and regards the virtual partial sequence starting at the latest data as the next partial sequence, as shown in Fig. 10. This enables the user and application to obtain the required data from the next partial sequence.

3.5 Approach to Multiple Interval Prediction

Feature quantity patterns express general features of partial sequences, but cannot determine the details. Thus, a cluster centroid cannot express the details of partial sequences associated with a cluster.

In addition, the predicted partial sequence length depends on the length of the cluster centroid used by prediction. Therefore, this depends on the length assumed by the classification system used in detection trends, because it is necessary to extract feature quantity patterns from partial sequences divided with the same interval as the interval of partial sequences to construct an accurate classification system.

According to this, the details on predicted partial sequences may differ from actual measured data. More specifically, this difference increases when the method predicts short partial sequences by using cluster correlation tables and cluster transition tables based on the classification system for long partial sequences. For example, the details on a centroid, which short partial sequences created, are lacking in long sequences, as shown in Fig. 11.

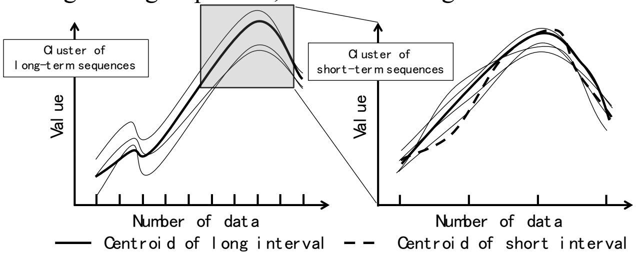

Figure 11: Data Lacking by Long-term Partial Sequences

Therefore, our method creates classification systems and tables with various partial sequence lengths. It selects the classifier and table in accordance with the time when a user and application want to obtain data. For example, if data for the near future are required, the method uses the tables based on short intervals. This enables the user and application to obtain accurate predicted data.

3.6 Using Past Measurements

Our method detects correlations in data streams from cluster correlation table and cluster transition table. Because these tables require a certain amount of past data to detect of correlation accurately, our method never detects correlations until these tables store enough past data. Therefore, we use the clustering results that are used for building a classifier to initialize two tables. We think this attempt improves the accuracy of prediction when real time trend detection is launched.

If unique measurements cannot be detected accurately by using classifier appearance, the accuracy of prediction decreases. This case may often happen in data streams treating seasonal data, such as measurements of temperature sensors in natural environments. Therefore, we suppose the classifier and two tables will be updated with a certain time.

4 EXPERIMENTS AND DISCUSSION

4.1 Evaluation of Clustering Algorithm

We carried out basic experiments to verify the efficacy of our method. We prepared sample data for the experiments and built classification system with various lengths.

This experiment used the stored data measured by a sensor unit used for farming [11] located in Fukushima Prefecture. This sensor unit measured the temperature, humidity, and illuminance from November 25, 2010 to February15, 2011. We only used 49 days worth of data from this span, when the sensor unit measured for 20 h between 0:00 to 20:00. The lengths of partial sequences were {20, 10, 4, 1}.

We defined the cluster evaluation value for prediction *CE* as the expectation value for prediction error obtained with Eqs. (3) and (4). Variable k is the number of clusters in these equations, and variable m is the number of partial sequences. In addition, variable c_{kj} means the *j* th data of the cluster centroid in cluster *k*, and x means one of the data.

Table 3: Accuracy of Each Classifier

Sensor	20 h			10 h			4 h			1 h		
	J48	MP	BN	J48	MP	BN	J48	MP	BN	J48	MP	BN
Temperature	95.9	92.3	97.8	95.9	97.6	90.2	89.8	84.8	71.9	80.6	60.2	57.1
Humidity	100	100	90	97.9	97.6	92.0	92.9	90.2	85.5	90.0	49.7	65.6
Illuminance	97.5	93.9	90.2	100	92.3	77.0	87.2	84.2	77.4	55.1	76.1	76.0

Variable x_{kij} means the *j* th data of the *i* th partial sequence at cluster k.

$$Dki = \left\{ \sum_{j=1}^{m} \frac{100}{(max - min)} \times \sqrt{(c_{kj} - x_{kij})^2} \right\} \Big/ m \quad (3)$$

$$CE_k = \left\{ \sum_{i=1}^{n} D_{ki} \right\} /n \quad (4)$$

Variable D_{ki} means the average of all deltas with all data of the cluster centroid. This variable is the delta of the *i* th partial sequence in cluster *k*. CE_k is the average of all deltas of partial sequences in cluster *k*. Variable *max* and *min* are the maximum and minimum values according to sensor performance. A delta is expressed as a percentage.

Tables 1 and 2 summarize the results obtained from the experiment. Table 1 lists the numbers of clusters and cluster evaluation values for prediction. Table 2 lists the expected number of partial sequences in each cluster. *CE* reached below 10 percent in most lengths of partial sequences. This means the error between actual data and predicted data will reach below at least 10 percent. For example, the error will reach bellow 4C in the temperature sensor. We assumed this error would be permissible for prediction in most cases. The expected numbers of partial sequences were near three in most lengths in Table 2. This is a particularly accurate result for our clustering algorithm that classifies the source into two clusters as recursively as possible. The numbers of clusters increase as intervals shorten. This means unique types of partial sequences increase in inverse relation to decreasing interval lengths. Therefore, we attributed these results to our clustering algorithm being able to detect unique types.

In both tables, the illuminance sensor's results in cases where the interval length is 4 or 1 h differ from the other results. These accented results mean trend-detection failed in these cases. We assumed our method failed because the feature quantities used in it were not compatible with the illuminance data in these interval lengths. Therefore, the method must correctly select feature quantities in accordance with the types of sensors and interval lengths.

4.2 Evaluation of Classifier Algorithm

We carried out a basic experiment on a classifier to determine an adequate algorithm for constructing it. Three classifiers were constructed in the experiment by using C4.5 and Multilayer Perceptron and Bayesian Networks, and we conducted 10-fold cross-validation to obtain the accuracy of the classifier. All algorithms were implemented in Weka. Multilayer Perceptron is an algorithm for building neural networks. Accuracy was expressed by the percentage of correct classifications. The experiment used the clustering results at each interval of partial sequences.

Table 3 summarizes the accuracy of each classifier where J48 is much more accurate than the others. Therefore, J48 was considered to be the best algorithm. In addition, the best accuracy was over 90% at 20 and 10 h. However, at 4 and 1 h, accuracy worsened. Most accuracy at 1 h especially fell below 80%. We assume the algorithms failed to construct classifiers in these cases. The algorithms failed because of our clustering algorithm, which could not consider the threshold used by the classifier, because our clustering algorithm executed classification in accordance with the Euclidean distance of the feature quantity pattern made from each partial sequence. We assumed the clustering algorithm would be ruled unsuitable as an algorithm for constructing classifiers by using a threshold. The clustering algorithm needs to be used by considering threshold of feature quantity to succeed in constructing classifiers.

5 CONCLUSION

This paper proposed a method for detecting the "timing correlation" between multiple data streams by using information on the appearance of trends. The basic experiments demonstrated the efficacy of the method for prediction using clusters created from the detection of trends. However, we found problems with the clustering method and feature quantities in clustering. It is necessary to do further research on adequate feature quantities for each sensor and intervals to find a new clustering method. In addition, we will verify the efficacy of our method by additional experiments. We will estimate clusters of past measured data by using the proposed method. The accuracy will be given by matching estimated clusters to actual clusters detected by classifier. If this experiment shows good accuracy, we will implement the system predicts data stream in real time, which using our method, in order to verify the efficacy against a single data stream prediction, where multiple data streams exist.

REFERENCES

[1] Y. Sakurai, ``Stream mining technologies for time-series data,'' IPSJ Magazine, Vol.47, No.7, pp.755-761 (2006). (*in Japanese*)

[2] H. Arimura and T. Kida, ``Mining technologies for data stream,'' IPSJ Magazine, Vol.46, No.1, pp.4-11 (2005). (*in Japanese*)

[3] Y. Fujiwara, Y. Sakurai and M. Yamamuro, ``A search method for multiple data streams,'' Database Society of Japan Letters, Vol.4, No. 4, pp.13-16 (2006).

[4] K. Sugawara, Y. Shiraishi and O. Takahashi, ``Estimation of the user action and static object using

3-Axis Acceleration sensor and vibration motor based on PWM control for cell-phone,'' Collection of papers of DPS Workshop 2009, pp.25-30 (2009). (*in Japanese*)

[5] H, Kawashima, M. Toyama and M. Imai, Y. Anzai, ``Retrieval of similar sequences with waveform features,'' technical report of IECIE SIG-DE, Vol.102, No.209, pp.125-128 (2002). (*in Japanese*)

[6] M. Toyoda, Y. Sakurai and T. Ichikawa, ``Stream matching based on Dynamic Programming,'' IPSJ SIG Technical Report DBS, Vol.2008, No.88, pp.277-282 (2008). (*in Japanese*)

[7] S. Papadimitriou, A. Brockwell and C. Faloutsos, ``Adaptive, hands-off stream mining,'' Proceedings of the 29th International Conference on VLDB, pp.560-571 (2003).

[8] Y. Sakurai, S. Papadimitriou and C. Faloutsos, ``BRAID: stream mining through group lag correlations,'' Proceedings of the 2005 ACM SIGMOD, pp.599-610 (2005).

[9] Y. Zhu and D. Shasha, ``StatStream: statistical monitoring of thousands of data streams in real time,'' Proceedings of the 28th International Conference on VLDB, pp.358-369 (2002).

[10] S. Onishi, Y. Sakai, N. Yamaguchi and T. Shiraishi, ``e-kakashi project,'' HTML, Available at "http://project.e-kakashi.com/" (2010). (*in Japanese*)

[11] Machine Learning Group a University of Waikato, ``Weka 3 - Data Mining with Open Source machine Learning Software in Java,'' HTML, Available at http://www.cs.waikato.ac.nz/ml/weka/

(Received February 27, 2012)
(Revised November 23, 2012)

Kenji Arai received B.E. and M.S. degrees from Future University Hakodate in 2010 and 2012. His research interests include sensor database system and data mining. He currently works at KDDI corporation in Japan.

Yoh Shiraishi received doctor's degree from Keio University in 2004. He is currently an associate professor at the Department of Media Architecture, School of Systems Information Science, Future University Hakodate Japan. His research interests include database, mobile sensing and ubiquitous computing. He is a member of IPSJ, IEICE, GISA and ACM.

Osamu Takahashi received master's degree from Hokkaido University in 1975. He is currently a professor at the Department of System Information Science at Future University Hakodate, Japan. His research interest includes ad-hoc net-work, network security, and mobile computing. He is a member of IEEE, IEICE, and IPSJ.

[Practical Paper] Inspection Method of Privacy Utilization for Rule-based Mobile Applications

Chiaki Doi†, Tomohiro Nakagawa†, Ken Ohta†, and Hiroshi Inamura†

†Research Laboratories, NTT DOCOMO, Japan
chiaki.doi.tf@nttdocomo.com

Abstract - As smartphones that offer the open development environment prevail, the number of applications is growing exponentially across the globe. Among them, context-aware mobile applications are increasing because various sensors are available to catch the user's context changes.

In order to create context-aware mobile applications, we adopt the ECA rule-based approach, which is intended for reactive applications. The application can perform functions without user's intervention, by leveraging context data such as terminal logs or sensor data as a trigger to perform predefined actions. Because context data can involve privacy data, we must ease privacy concerns before these applications will be widely accepted.

The objective of this research is to realize a method that determines how privacy data is utilized in context-aware rule-based mobile applications. The method outputs a privacy report, which is shown to users so that they can confirm if the application is acceptable or not in terms of privacy. There are three requirements that prescribe how privacy data utilization is to be described in the report.

The challenge of inspecting rule-based applications is to analyze the information flow on the implicit chain formed by independent rules. Such an implicit chain exists because there are cases in which the firing of one rule depends on another, even if the rule doesn't explicitly refer to the other.

Our privacy data inspection method is composed of a chain analysis process, a filtering process, and a summarization process. The proposed method satisfies the three requirements for privacy reporting, as well as accuracy and conciseness.

In order to evaluate the chain analysis and filtering processes, we create an evaluation model composed of two rules, which is the minimum rule chain unit. We confirm that neither false positives nor false-negatives are detected in evaluations that uses all 400 combinations of the rules generated from the model. We also confirm that the summarization process is effective in creating concise privacy reports. From this result, it is concluded that the proposed method works correctly for rule-based applications composed of straight chain rules.

Keywords: ECA rule, Information flow, Privacy data

1 INTRODUCTION

Due to the increasing number of mobile phone functions now open to application developers, it is becoming more attractive to create context-aware mobile applications. These applications leverage terminal logs such as user's location information or sensor data and behave autonomously by taking the user's context into account. For example, BreadCrumbz [1], Locale [2], and Nakamap [3] use such information to navigate people to specified places.

While the application market is expanding rapidly, concern is growing as to the security of using 3rd party applications. In recent studies of the Android platform, it has been pointed out that the Android framework faces high-risk threats[1][2], and that it is essential to determine how well applications can guarantee the Android user's security and privacy[3][9].

In order to create context-aware applications, we adopt the ECA rule, which is a suitable way to describe an application that automatically executes various functions depending on the user's situation [16].

Based on this ECA-rule based approach, we tackle the challenge of application inspection. Because the applications utilize the user's privacy data such as terminal log and sensor data, they suffer the risk of disclosing privacy information. For example, the application may directly pass the user's privacy data such as location information to another user. As another example, the application may indirectly convey the user's context to other users in the case that the user's context is utilized as the trigger to execute notification function to the other users.

In this paper, we propose a method to inspect how privacy information is utilized in ECA rule-based mobile applications. The proposed method is composed of a chain analysis process, a filtering process, and a summarization process. After these three processes conclude, the result, a privacy report, is shown to the user, the user can then confirm whether the application exhibits any harmful behavior or not before installing it.

In order to evaluate the chain analysis and the filtering processes, we created an evaluation model composed of two rules, which is the minimum rule chain unit. An evaluation result based on evaluations of all 400 combinations of the rules generated from the model shows that neither false positives nor false-negatives are detected. Thus it is confirmed that the proposed method works correctly for straight chain rules, and thus covers simple applications.

In addition, we evaluate the summarization process, and the result shows that the amount of privacy data is nearly halved in the privacy report while keeping the information

[1] BreadCrumbz http://www.bcrumbz.com/
[2] Locale http://www.twofortyfouram.com/
[3] Nakamap http://www.appbrain.com/app/nakamap-where-are-you-now/com.kayac.nakamap

needed by the user for decision making.

The paper is organized as follows. Section 2 describes the ECA rule-based application and discusses related work; it focuses on existing static analysis techniques and tools. Section 3 describes three requirements for the privacy report to be accurate and concise. Section 4 introduces our proposed method. In this section we explain how to analyze the chain of the rules, how to filter unnecessary rule chains from the candidates, and how to summarize similar items to make the privacy report concise. Section 5 describes the evaluation model for straight rule chains, i.e. no branches or loops are present. Also, the result of an evaluation of the summarization method is shown in this section. Section 6 concludes with a summary and an outlook on future work.

2 ASSUMPTION AND RELATED WORK

2.1 ECA rule-based applications

In this study we adopt the ECA rule[4][5][17][18] to describe context-aware mobile applications, which autonomously execute functions depending on the user's context. The ECA rule is composed of an event, a condition, and an action. The event triggers the processing of the rule. After the event occurs, the specified condition is checked. Only when the condition is satisfied is the action implemented. The reason why we adopt the ECA rule is that the execution of the rule is conducted without the user's explicit intervention, which is a suitable attribute for context-aware applications.

A rule-based application is composed of a set of ECA rules, which are described in XML. Each ECA rule in the application works independently, and has a different trigger as defined by the event. Thus, the rule has atomicity in that its logic is self-contained. Due to rule atomicity, a ECA-rule based application has flexibility in terms of execution, which is an additional reason why we adopt the ECA rule. For example, previous research mentioned that an ECA-based application can be customized by adding or deleting rules [6]. Also, there is research on dynamic changing rules at runtime in a transparent manner [7]. We consider that part of the ECA rules will, in the future, be transferred between a terminal and a server to realize dynamic load balancing.

In order to describe various types of mobile applications, we define several tags for the event and the condition. These tags are utilized to catch the change in the user's context through the user's operation log or sensor data. For example, an *OCCUR* tag is defined to catch the generation of a specified terminal log. As another example, a *SUM* tag is defined to evaluate if total number of terminal log generation is more or less than a specified value.

Also, various kinds of terminal logs are available as the evaluation target for these tags. There are several kinds of logs such as screen light-up or application start-up, which indicate the user's operation of the mobile phone. In addition, there are other kinds of logs such as location information or pedometer data, which indicate the user's behavior. By combining the aforementioned tags and these terminal logs, you can describe different kinds of context-aware applications.

In addition, we define a user-defined event as a kind of log. The objective of the user defined-event is to gather the user's context from their pressing a button on a dialog. Suppose the user-defined event *BEING_TIRED* is bound to a dialog button. When the button is pressed, the corresponding user-defined event is issued and recorded as user's operation log. Afterwards, the log of *BEING_TIRED* can be utilized in the same way as the normal kind of log from the event or condition by using tags such as *OCCUR* or *SUM*.

As an example of ECA-rule based applications, let's take an information distribution application which provides tourists with information depending on their location and pedometer data as captured by mobile phones. Figure 1 shows an excerpt of an ECA rule from the application. The rule defines that the event is fired when the user's location is within 1 km from Venice, which is located at latitude 45.434336 north and longitude 12.338784 east. Then, as defined in the condition, the pedometer data is checked to determine if it exceeds 5000 steps. If yes, as defined in the action, an implicit Intent is issued by using the Intent system of Android, and the browser accesses the specified URL of a web service which distributes tourist information. When the browser accesses the 3rd party's server, it transmits the user's location and pedometer data to obtain the user's context, which is utilized to personalize the delivered information.

This sample rule shows that the user's location and pedometer value is sent to the information provider. Thus, the user will want to know what kind of privacy data is being utilized and how the data is being conveyed to 3rd parties, before using the application.

```
<rule id="1">
 <event>
  <center lat="45.434336" lon="12.338784"
     kind="LOCATION_INFORMATION">
   <less_than>1000</less_than>
  </center>
 </event>
 <condition>
  <sum kind="PEDOMETER" more_than="5000">
  </sum>
 </condition>
  <action>
   <coordinate intentType="implicit"
     data="http://URL1?location=Venice&pedometer=
5000"
       action="android.intent.action.VIEW"/>
 </action>
</rule>
```

Figure 1: An example of ECA rule-based application

2.2 Rule chain by dependency between rules

As we have seen in the previous section, an ECA rule-based application is described as an aggregate of independent rules. In other words, no rule explicitly refers to any other rule.

In spite of rule independency, there is an implicit dependency between the rules in specific situations. Such a situation occurs when the execution of a specific rule's action is a necessary condition for firing another rule's event. If there is such dependency between the rules, user's privacy data may indirectly flow out of the terminal through the continuous execution of these rules. Supporse there are two rules. The first rule's event has a SCREEN_ON tag and action has a SEND_USER_DEFINED_EVENT tag. The second rule's event has a RECEIVE_USER_DEFINED_EVENT tag and action has a SEND E-MAIL tag Rule. The second rule sent a email to specified address after the first rule fired after user touches screen. In this case, email recipent can know that the user operated the terminal. In order to clarify an output path of privacy data, we need to analyze the dependency between rules.

There are two cases of such rule dependency, which is called 'rule chain' hereafter.

In the first case, the first rule's action has a *LOG START* tag and the second rule has any kind of event tag which utilizes the same kind of log specified in the first rule. The *LOG START* tag means to start recording a kind of log. In the second case, the second rule's event has an *OCCUR* tag, which means to catch the generation of a specified log type, the second rule is fired only after the first rule is executed.

The second case of the rule chain is that the first rule's action has a *BUTTON* tag, which issues a user-defined event, and the second rule's event has any kind of event tag which utilizes the same user-defined event specified in the first rule. If the second rule's event has an *OCCUR* tag that is defined to count the number of the same user-defined event specified in the first rule, there is a chain between those rules.

2.3 Related work

For the purpose of tracking how sensitive data is handled by an application, some studies attempt to track the data at runtime[8][9]. In [8], sensitive data such as passwords or credit card numbers are tracked by simulating the whole system. TaintDroid[9] realizes a light-weight system-wide tracking system for mobile phones.

While these tracking systems have the same purpose, analyzing the flow of sensitive data, they use dynamic analysis, which assumes that the target application can be run in an emulation environment or a real system.

Our approach is to use a static analysis approach, which can analyze the information flow without executing the target application. By using static analysis, the user can notice the malformed behavior of an application before installing it.

There are several techniques for the static analysis of information flow that annotate programs handling confidential information [10][11]. Our proposal avoids such annotation. All of the privacy data is predefined as confidential information, and the flow of the privacy data is exhaustively analyzed. Then, whether the privacy data should be reported or not is judged by how they are utilized.

There are some techniques that offer the static analysis of data or programs without annotation [12][13][14]. These techniques are intended to verify the query data[12][13] or sequential execution of the code, which follows an explicit control[14].

The difference between our proposed method and existing static analysis techniques is we analyze the implicit chain formed by independent rules. This study is intended for applications that consist of independent rules. However, as described in Section 2.2, the rules can form implicit chains. Therefore, it is necessary to consider the dependence described in Section 2.2. To analyze an implicit rule chain, it is necessary to analyze the pathways of information by considering the semantic relationships between rules.

From the viewpoint of analyzing rule-based applications, a previous study described the verification of an application's convergence[15]. However, its goal was to verify whether the application's running time would converge or not, and so cannot be applied to our problem.

Existing related tools are Permission Checker Security[4] and S2Permission Checker[5] to detect applications that might be spyware applications in Android phones. The difference between these tools and our method is that those tools merely show the possibility of anomalous behavior at a rough level of granularity. For example, these tools do not show the destination of the leaked information. On the contrary, our proposed privacy report provides users with a more detailed assessment of the behavior of the application as described in Section 3.

3 PRIVACY UTILIZATION INSPECTION

3.1 Requirements

After the application is inspected, the result will be presented to the user as a privacy report as shown in Fig. 2. The privacy report is meant to be shown before the user installs the application. The user can then judge whether or not it is acceptable to utilize the application by confirming how the user's privacy data is utilized.

In order to clarify the detailed behavior of the application, there are three requirements that the privacy report must satisfy. The motivation of detailing the privacy utilization as described below is that there are various terminal logs, and a subtle change in how they are utilized may change the mind of the user.

1. Show which kind of privacy data is utilized and to which destination the information is sent
2. Details the conditions under which the privacy data was created
3. Eliminate redundancy from the privacy report so that the user can understand it easily

[4] Permission Checker Security
http://www.appbrain.com/app/application-permission-checker/jp.ne.neko.freewing.PermissionChecker
[5] S2 Permission Checker
http://www.androidzoom.com/android_applications/tools/s2-permission-checker_luge.html

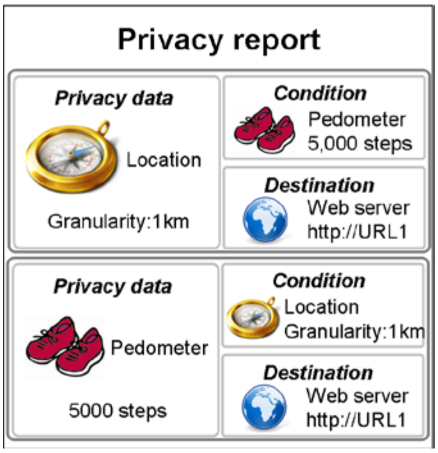

Figure 2: An example of the privacy report

Requirement 1 specifies that the privacy data such as user's location and pedometer data should be displayed jointly with the destination. The destination is specified for every notification method that uses the action. We defined eight kinds of actions such as e-mail transmission, Intent invocation, and location sharing. Existing tools specify only the privacy information that could be output. The addition of the destination makes the privacy report much more useful.

Requirement 2 specifies that the condition under which the action is taken should be included in the privacy report. We consider that the conditions may influence the user's judgment on whether or not to utilize the application. This is because the user should weigh the benefits and the privacy risks of the application in making the judgment. For example, suppose it is unacceptable for user A to share location information anytime and anywhere with person B. By limiting the time to share the location to around the time of rendezvous with B, the same application can turn to be an attractive one for the same user A.

Requirement 3 specifies that the privacy data described in the privacy report should be concise, eliminating items of the same sort. This is necessary to make the report readable.

3.2 Proposed inspection method

By adopting the ECA rule-based approach, we can describe a context-aware application flexibly by combining various kinds of tags and terminal logs. We designed an application programming language composed of seven events, three conditions, and eleven actions. In addition, 54 terminal logs can be utilized from the event and condition tags.

In order to generate accurate privacy reports, we need to clarify the implicit rule chain formed by any combination of the rules freely defined by utilizing these tags and logs. If there was any false recognition or oversight of a rule chain, the privacy report might contain false positives or false negatives.

In this section and the following subsections, we describe the proposed method, which is composed of the three processes shown in Fig. 3. The proposed method processes the ECA rules and generates a privacy report as its output.

The chain analysis process and second filtering process are designed to satisfy requirements 1 and 2, by clarifying the sequence in which the privacy data is utilized. The summarization process is designed to satisfy requirement 3, by aggregating the same kind of items in the privacy report.

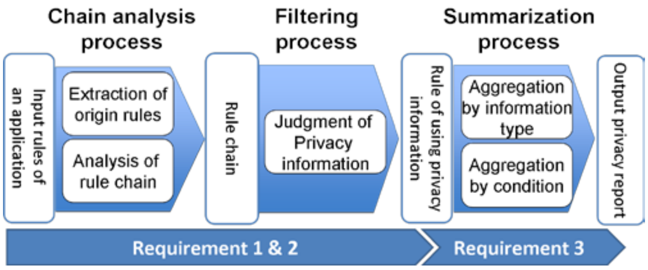

Figure 3: Processes of the proposed method

3.2.1. Chain analysis process

The rule chain analyzed in this process starts from an origin rule and the chain is formed by the dependency of the rules as described in Section 2.2.

The origin rule is defined as a source of the privacy data utilization. In other words, it is the rule that utilizes the user's privacy data in its event, condition, or action tags. For example, a rule that includes *OCCUR* tag or a *SUM* tag is regarded as an origin rule. On the other hand, if a tag doesn't utilize any privacy data, a rule that uses only that tag is not an origin rule. For example, *TIME* tag is fired at the designated time regardless of the user's situation. That's why a rule that includes *TIME* tag is not an origin rule.

Note that the origin rule is defined by not only the event tag, but also the condition and action tags. The reasons are as follows. The condition tag has a function of judgment by taking the user's context data as a parameter, just as the event tag does. The action tag doesn't have such function, as it either directly or indirectly transmits the user's privacy data. An example of direct transmission is a *LOCATION* tag, which has the function of sending location information to a server to share it with designated agents/persons. An example of indirect transmission is a *BUTTON* tag, which has the function of issuing a user-defined event. In this case, the user's button operation can be transmitted via a rule chain.

Figure 4 shows a flowchart of the chain analysis process. This procedure is performed in two steps.

Table 1: Example of privacy data in an origin rule

Category	Example of privacy data
User data	Phonebook entry added/modified
	Schedule added/deleted/modified
Terminal state	Battery power changed
	Silent mode set/canceled
Operation log	Screen light on/off
	Button operation
Behavioral information	User's location
	Pedometer data

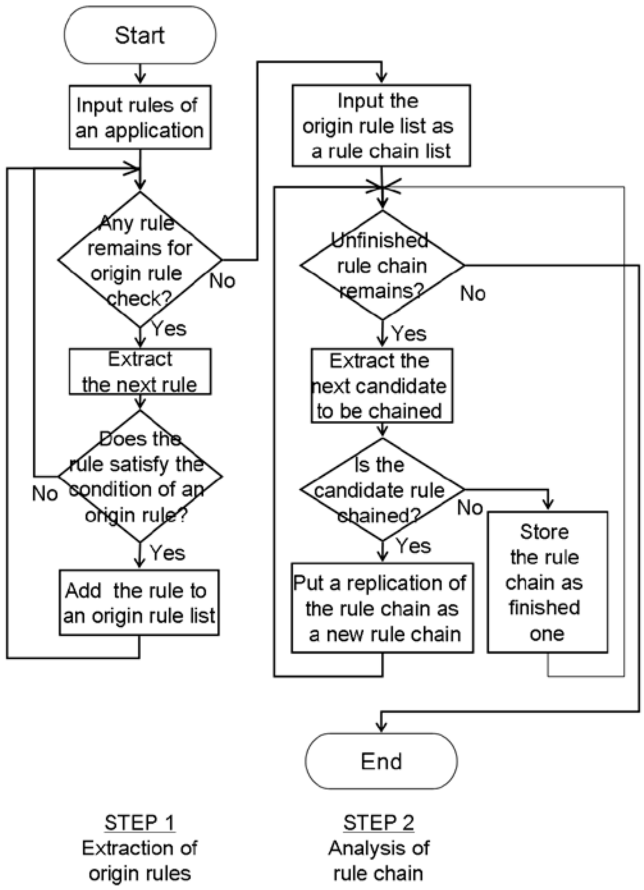

Figure 4: Flowchart of chain analysis process

The first step is to extract all origin rules, which is realized by pattern matching of the tags that utilize privacy data. There are four classes of privacy data: user data, terminal state, operation log, and behavioral information (see Table 1). User data is defined as data input by a user, such as schedule information or phonebook entries. Terminal state is defined as data that shows the status of the mobile phone, such as remaining battery power or silent mode. These data can be used to infer the current state of the user's situation. Operation is defined as the record of terminal operation by a user, such as screen light on and off, or pressing a button displayed on a dialog. The operation log makes it possible to determine whether there is a user's operation or not. Behavioral information is defined as data that shows user's activity in the real world, such as user's location or pedometer count. With this information, the system can infer where and what users are doing. These categories are utilized in the filtering process, as described in the next subsection.

The second step is to determine the rule chains starting from the origin rules extracted in the first step. This determines the pathway on which the user's privacy data is output. To extract all rule chains, whenever a new branch is found, the original rule chain is replicated. Loops are avoided by excluding rules that already exist in the rule chain.

The tricky part of the second step is that in some exceptional cases we need to define a rule chain as invalid, considering the semantics of the tags. For example, because *LOCATION* tag directly outputs user's location information by itself, a rule with the tag must be regarded as a finished rule chain, and so cannot be connected to any other rule.

3.2.2. Filtering process

At the end of the rule chain analysis, we get a set of rule chains, which are filtered to extract the items essential for the privacy report. The motivation of the filtering process is to remove trivial items from the report. Even if a rule chain starts from privacy data utilization, the privacy data need not be present in the privacy report as long as the data is utilized internally or in a normal way.

For filtering the privacy data, we focused on the relationship between privacy data in the origin rules and how the data is finally utilized in end rules. The end rule is defined as a rule that includes an action tag with an output function.

Table 2: Criteria for privacy data filtering

	Utilization purpose		
	Network output	Display output	Internal processing
User data	○	-	-
Terminal state	○	-	-
Operation log	○	○	-
Behavioral information	○	○	-

○ : Described in the privacy report
- : Omitted from the privacy report

The criteria for the filtering process are shown in Table 2. The intersection of the categories of privacy data and utilization purpose shows whether the privacy data should be described in the privacy report or not. The utilization purpose is categorized as network output, display output, and internal processing. In our filtering algorithm, all privacy data that is output via the network is reported to the user. On the other hand, only operation log and behavioral information are reported when these privacy data are output via the mobile's display. This is because user data and terminal data can be displayed for other applications, and these cases are regarded as relatively insignificant.

Just as in the case of the chain analysis process, we need to be careful of slight differences in the semantics of tags in the filtering process. We need to disable an end tag when the origin rule doesn't output user's context in a single rule. For example, a *BUTTON* tag is an origin tag which shows user's button operation. Even a rule including the *BUTTON* tag has also an action tag with output function, the button operation should be omitted from the privacy report. This is because the *BUTTON* tag is a part of an action tag . Only if the rule with the *BUTTON* tag is chained to other rules and the chained rule includes an output function, should the button operation be described in the privacy report.

3.2.3. Summarization process

In the summarization process, we eliminate the redundancy of the information presented to the user to satisfy requirement 3. This process aggregates similar rule chains by classifying them into the same category. These rule chains are aggregated when the same operation and behavior information are output by the same function.

We implemented two methods for the summarization process. Table 3 shows the items to be considered in the summarization process. The first method aggregates the rule chains by considering the conditions of privacy information. The second method does not consider these conditions.

Table 3: Items considered in the summarization process

	Origin rule		End rule
	Privacy data	Condition	
Method 1	○	○	○
Method 2	○	—	○

○ : Considered , — : Not Considered

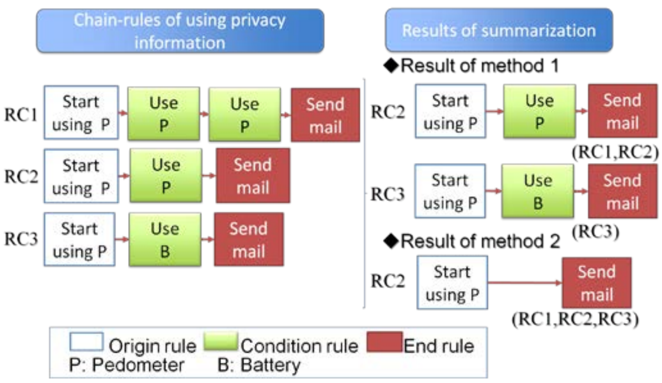

Figure 5: Operation of summarization process

Figure 5 shows the operation of the summarization process.

In method 1, judgment of whether the pedometer data meets the condition is performed for the second and the third rule. The second and third rules set the same condition; therefore, these rules are placed in the same category. RC1and RC2 are placed in one category if they have the same combination of privacy information and condition. As a result of summarization process, RC1 is aggregated into RC2 because RC2 has fewer rules than RC1. RC2 is output as the result of the summarization process. RC3 has different condition in the second rule, so RC3 is also output as an independent result.

In method 2, rule chains are aggregated without consideration of the conditions. Only the origin rule and the end rule are used for aggregation, so the amount of information output by summarization is reduced. The result of applying this filtering process to RC1, CR2 and RC3, is that RC1 and RC3 are aggregated into RC2 which has common origin rules and end rules.

The summarization process is stronger with method 2 than with method 1. The latter can show details that include the conditions of using privacy information.

4 EVALUATION

4.1 Filtering and chain analysis processes

In order to evaluate the filtering process and the chain analysis process, we created an evaluation model composed of two rules, which is the minimum rule chain unit. Figure 6 shows the evaluation model proposed here. The rules in the evaluation model are composed of only event and action, i.e. they do not include any condition. The reason for this omission is that event and condition have the same mechanism for extracting privacy data.

For the sake of model completeness, the tags utilized in the evaluation model are selected so that they encompass all the categories that can impact the proposed processes. The result of the filtering process is influenced by the category; therefore we chose a representative tag from each category. We neglect the privacy data utilized in the tag because the difference in privacy data has no influence on the result of the chain analysis process.

Table 4 shows the tags used in this model. The event in the first rule includes four tags to cover all the categories of the privacy data which we defined in Table 1. The action in the first rule includes two tags which chain to other tags and three tags that output tags which include privacy information. The event in the third rule includes three tags which chained to other tags and two tags are chained to other tags. The action in the fourth rule includes three tags which output privacy information and a tag which finishes recording the log.

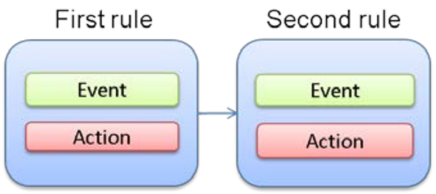

Figure 6: Evaluation model

Our evaluation model effectively reduces the number of samples needed for the evaluation. Our model needs only 400 samples generated by the combinations of the representative tags. Without our model, 1,171,800 combinations would have to be evaluated.

An evaluation using all 400 combinations of the rules generated from the model confirms that neither false positives nor false negatives were detected. We covered applications which chain linearly in this evaluation model. In other words, applications that include branches and loops are not covered. It is a future work to build a general evaluation model to cover such applications.

Table 4: Tags in evaluation model

Rule		Function	Tags used in the evaluation model
The first rule	Event	Tags which utilize user's privacy data	<sum kind= "21">
			<sum kind= "43">
			<sum kind= "40">
			<sum kind= "28">
	Action	Tags which chain to other tags	Send user-defined event
			Start to record the privacy data
		Tags which output privacy information	<location>
			<dialog>
			<timer>
The second rule	Event	Tags which chain to other tags	receive user-defined event
			Utilize the privacy data
		Tags which don't chain to other tags	<occur kind="43">
			<occur kind="40">
			<occur kind="28">
	Action	Tags which don't chain	<log stop>
		Tags which output privacy information	<location>
			<dialog>
			<timer>

4.2 Summarization process

Our evaluation of the summarization process used the 18 kinds of tags shown in Table 4. We generated 25 applications, each of which had a set of five rules. The rule sets were generated by randomly choosing tag from among the candidate tag group with equal probability. The average number of the privacy data in the privacy report was 25 for the original data before the summarization process. After applying our summarization method, the number of the privacy data was reduced to 13 for method 1, and 11 for method 2. Thus, we confirmed that the summarization process is effective in reducing the number of private data, making it possible for users to confirm the privacy report more easily.

4.3 Processing time

We used Java to implement our proposed method. Table 5 shows the execution environment in which we evaluated our proposal.

Table 5: Execution environment

Execution environment	
CPU	**IntelCore2 6400@2.13GHz**
OS	**Windows XP Professional SP3**
RAM	**2GB**

Figure 7 shows the average processing time of the proposed method. The total processing time is the time required for filtering process, chain analysis process, and summarization process. The test data were the 25 applications utilized in the evaluation of the summarization process.

The average total processing time of the proposed method is 41 ms for method 1, and 37 ms for method 2. The result ranged from 32 ms to 44 ms for methods 1 and 2. This result confirms that either method 1 or 2 can be selected depending on the user's requirement.

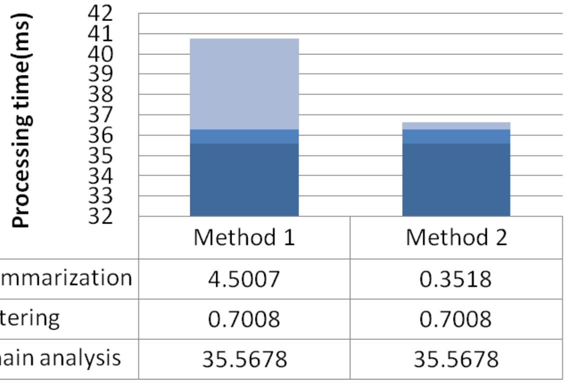

Figure 7: Processing time of the proposed method

5 CONCLUSION

We proposed an application inspection method for rule-based applications that is composed of a chain analysis process, a filtering process, and a summarization process. We confirmed the validity of the proposed method by using an evaluation model with minimum rule chain configuration. The evaluation results show that the proposed method works correctly for applications that are composed of straight rule chains. We also evaluated two summarization methods and confirmed that our method cut the amount of information present on the privacy report.

As a future work, we plan to evaluate the chain analysis and filtering processes by a more generalized model. In such a model, we need to include branches and loops of the rule chain to cover complicated applications. It is also a future work to improve the clarity of the privacy report in case there is many kinds and granularity of privacy data as well as several ways of notification to other users. As the severity of privacy risk depends on the combination of these elements, it is a challenge how to create concise privacy report that includes required alert a user.

REFERENCES

[1] A. Shabtai, ``Google Android: A Comprehensive Security Assessment,'' IEEE Security & Privacy Magazine, pp. 35-44 (2010).

[2] W. Enck, ``Understanding Android Security, Security Assessment,'' IEEE Security & Privacy Magazine, pp. 50-57 (2009).

[3] M. Butler, ``Android: Changing the Mobile Landscape,'' IEEE Pervasive Computing, Vol. 10, No. 1, pp. 4-7 (2011).

[4] J. Bailey, ``An event-condition-action language for

XML," Proceeding of the 11th international conference on World Wide Web, pp. 486-495 (2002).
[5] G. Papamarkos, ``Event-condition-action rule languages for the semantic web, " Workshop on Semantic Web and Databases, pp. 855-864 (2003).
[6] M. Miyama, ``An event-driven navigation platform for wearable computing environments," Proceeding of the 9th IEEE International Symposium on Wearable Computers, pp. 100-107 (2005).
[7] S. Chakravarthy, ``Dynamic Programming Environment for Active Rules," The 7th International Baltic Conference on Database and Information Systems, pp. 3-16 (2006).
[8] J. Chow, ``Understanding Data Lifetime via Whole System Simulation," USENIX Security Symposium, pp. 321-336 (2004).
[9] W. Enck, ``TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," Proceeding of the 9th USENIX Symposium on Operating Systems Design and Implementation, pp. 1-15 (2010).
[10] E. Dorothy, ``A lattice model of secure information flow," Communications of the ACM, Vol.19, No.5, pp.236-243 (1976).
[11] A. C. Myers, ``JFlow: Practical Mostly-Static Information Flow Control," Proceeding of the 26th ACM Symposium on Principles of Programming Languages, pp. 228-241 (1999).
[12] P. Geneves, ``XML Reasoning Made Practical," The 26th International Conference on Data Engineering, pp. 1169-1172 (2010).
[13] S. Boettecher, ``Finding the Leak: A Privacy Audit System for Sensitive XML Databases," Proceeding of the 22nd International Conference on Data Engineering Workshops, p. 100 (2006).
[14] C. Kirkegaard, ``Static Analysis of XML Transformation in Java," IEEE Transaction on Software Engineering, Vol.30, No.3, pp181-192 (2004).
[15] A. Cheng, ``Fast Static Analysis of Real-Time Rule-Based Systems to Verify Their Fixed Point Convergence," Proceedings of the 5th Annual Conference on Software Safety and Process Security, pp. 46-56(1990).
[16] P. Moore, B. Hu and M. Jackson, ``Rule Strategies for Intelligent Context-Aware Systems," Proceedings of the 2011 International Conference on Complex, Intelligent, and Software Intensive Systems, pp. 9-16 (2011).
[17] T. Hu and B. Li, ``Research on the Mechanism of Tourism Information Change Management Based on ECA Rules," Proceedings of the 2010 13th IEEE International Conference on Computational Science and Engineering, pp. 388-392, (2010).
[18] P. Moore, M. Jackson and B. Hu, ``Constraint Satisfaction in Intelligent Context-Aware Systems," Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems, pp 75-80, (2010).

(Received February 13, 2012)
(Revised February 28, 2013)

Chiaki Doi joined NTT DOCOMO, Inc., in 2009. Her research interests include mobile computing, system security, and computer graphics. She received B.E. degree from Iwate Prefectural University and M.E degree from Keio University, Japan. She is a member of Information Processing Society of Japan.

Tomohiro Nakagawa is a research engineer at NTT DOCOMO, Inc. His research interests include mobile computing and distributed systems. He received his B.E. and M.E. from the University of Tokyo, Japan in 1998, and 2000, respectively. He is a member of IPSJ.

Ken Ohta received the BE, ME, and DE degrees from Shizuoka University, Japan in 1994, 1996, and 1998, respectively. In 1999, he joined NTT Mobile Communications Network Inc. (NTT DOCOMO). His research interests include mobile computing, distributed systems, and system security. He is a member of the Information Processing Society of Japan and of the Institute of Electronics, Information and Communication Engineers.

Hiroshi Inamura joined NTT DOCOMO, Inc. in 1999. His research interests include system research on mobile device and distributed system. Before DOCOMO, he was a research engineer in NTT labs since 1990.From 1994 to 1995, he was a visited researcher in the Department of Computer Science, Carnegie Mellon University. He received B.E., M.E. and D.E. degree in Keio University, Japan. He is a member of IPSJ, IEICE, and ACM.

[Practical Paper] LASP — a Learning Assistant System for Formal Proofs — and Its Application to Education

Kiyoyuki Miyazawa†, Kozo Okano†, and Shinji Kusumoto†

†Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita City, Osaka 565-0871, Japan

Abstract - The basis of formal techniques is mathematical logic. Especially, it is important to understand the concept of formal proofs. It is, however, difficult for novices to study formal proofs because of the rigorousness of the operations involved. To solve this problem, we propose and implement a prototype of a learning assistant tool for formal proofs called LASP. The purpose of LASP is to prevent the difficulties which occur when normal learners perform exercises by hand. The advantages of the proposed method are the following: 1. input support for long logical expressions; 2. the users are not required to perform copy and paste operations when they construct proofs; 3. hint features facilitate the construction of proofs; and 4. the proofs can be output as LaTeX files. Experimental results show that LASP avoids the drawbacks of conventional exercises.

Keywords: Logics, Formal Proof, Computer Aided Education

1 Introduction

The formal approach is a core technique of modern software development. There are mainly two approaches referred to as the formal approach: interactive theorem proof and the model-based approach, which includes model checking[2]. Well-known interactive theorem proof systems include Coq[6], Agda[7], and Isable/HOL, among other. These systems require the user to understand the logic of formal proofs. Model checking also requires the user to apply logic in order to represent properties on the target system. Other model-based approaches also require logic to describe constraints on models representing the target systems and software.

On the other hand, it is said that "Mathematical reasoning is intrinsic to both traditional engineering and software engineering, . . . Software engineers usually use discrete mathematics and logic in a declarative mode for specifying and verifying system behaviours and for analysing system features"[11]. This statement illustrates how it is important to understand mathematical logic in software engineering. Supporting the learning of mathematical logic can promote the learning of basic techniques of the formal approach. We can promote productivity in software development by teaching the formal approach.

A learning assistant system which we developed in this research supports learning formal proofs based on Hilbert's axiom system. As related works, Jon Barwise et al. have developed Turing's World[4] and Tarski's World[3], [5]. One can learn graphically Turing machine using Turing's World, and we can learn graphically the semantics of mathematical logic using Tarski's World. Tarski's World shows simple 3D computer graphic worlds in which geometric blocks of various kinds and sizes are distributed. Tarski's World gives you a first-order predicate sentence and a 3D figure and lets you decide whether the sentence is true or false for the provided figure. For example, a sentence "there exists a cube among the objects" is given and a user has to decide whether the sentence is true by looking at the figure. Three-dimensional views are sometimes used in computer-aided education. For example, paper[12] provides a 3D geometric construction tool specifically designed for mathematics, especially geometry, education. The tool is based on a mobile augmented reality (AR) system.

Another approach for computer-aided education is using a collaborative web-based environment, such as Moodle or WEB-CT. Several other approaches have been proposed, including paper [10], which provides a flexible learning scheme for selected pilot courses in engineering education. In such schemes, traditional lectures and written exercises are combined with separate Web-based learning resources.

In the case of logic-oriented education, MacLogic[9] supports the learning of Gentzen's natural deduction. Logic-Tutor[1] supports students as they are learning logic. It has a feature for analysing and reasoning a student's mistakes during the solving of logic exercises. It has been used in logic classes in the Department of Computer Science of the University of Sydney.

In this research, we target novices in logic and provide tools for them. Therefore, we do not consider that simply applying tools such as conventional theorem provers [6], [7] to education is suitable, although others have used a similar approach. For example, "Bringing Research Tools into the Classroom"[14] helps move computational tools used for research into the classroom. It successfully brings high- performance computing into modelling courses, builds software for both protein structure visualization and hydrological analysis of watersheds, and so on. A project of Buchberger [8] aims at supporting the entire process of mathematical theory exploration within one coherent logic and software system. This project uses formal logic and a computer-aided approach to help a learner to understand the core of mathematics.

While other related research mainly focuses on understanding semantics, here we focus on understanding formal proofs. For understanding the semantics of logic, an approach like Tarski's World, which utilizes 3D graphics, is useful. We, however, focus on formal proofs, in which 3D graphics may be of little help. In addition, our tool deals with Hilbert's axiom system because the class on mathematical logic held

Table 1: Axioms for First-order Logic

A1	$P \to (Q \to P)$	
A2	$(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$	
A3	$(\neg P \to \neg Q) \to ((\neg P \to Q) \to P)$	
A4	$\forall x\, T(x) \to T(t)$, where term t is free from variable x of $T(x)$.	
B1	$P, P \to Q \vdash Q$	(MP, modus ponens)
B2	$P \to Q \vdash P \to \forall x\, Q$, where term P is free from a free variable x	

in the Department of Informatics and Mathematical Science of Osaka University deals only with this system. As far as we know, no tools for learning Hilbert's formal proof exist. Therefore, we developed our tool and evaluated its effectiveness.

The paper is organised as follows. Section 2 presents a brief introduction of formal proofs. Sections 3 and 4 describe our tool and an experimental evaluation, respectively. Section 5 presents a discussion. Finally, Section 6 concludes our paper.

2 Formal Proof

A formal proof is a process which proves theorems by axioms and inference rules. An axiom is a major premise to derive a concrete logical expression instance without contradictions, while an inference rule is used to derive a new logical expression from proved ones. For example, modus ponens is an inference rule which derives Q from $P \to Q$ and P (where P and Q are arbitrary logical expressions).

An axiom system consists of axioms and inference rules, and the logical expressions proved by an axiom system are called theorems. Figure 1 shows an example of the formal proof which proves a theorem of $\vdash X \to X$.

Table 1 shows Hilbert's axioms and inference rules for first-order logic.

3 LASP, a Learning Assistant System for Formal Proofs

In this section, we will present our tool "Learning Assistant System for Proofs, (LASP)."

3.1 System Overview

We developed LASP to reduce problem practice time relative to writing proofs by hand. This tool is based on Hilbert's axiom system.

The drawbacks of practicing formal proof by hand are the following.

1. We may make a mistake in the writing of a long expression corresponding to an axiom, especially the correspondence between terms and variables.

2. We are often required to perform operations similar to copy and paste when constructing a formal proof.

To resolve the above problems, we define the goal as follows.

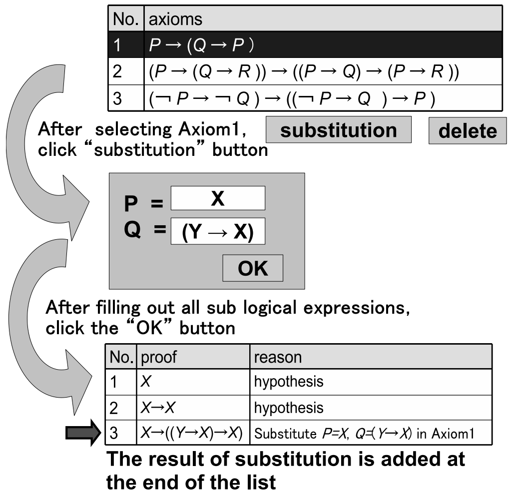

Figure 1: Substitution

1. LASP should have an interface to minimise the input of logical expressions.

2. We should not be required to perform operations like copy and paste.

LASP is implemented in Java with 8660 LOC. The system is available on the WEB[13].

We will next describe the specific features and interfaces of LASP.

3.2 Substitution Assistant

We implemented a substitution assistant feature in order to reduce time wasted during the proof; this feature also allows users to avoid careless written mistakes, such as inconsistencies between variables and terms, sentence structures, and variable name. Such mistakes also often occur when handwriting long and complex logical expressions. These mistakaes are unrelated to learning essential proofs. Instead, it is important to learn the thinking process of problem resolution in mathematical logic.

Let us consider the following example. Substitution of variables P, Q, and R of axiom A2 in Table 1 with the terms $X \to Y$, Z, and $((Y \to Z) \to X)$, respectively, yields the logical expression (1), which is obviously long and complex. This expression is also hard to read and write correctly.

$$\begin{aligned}((X \to Y) \;&\to\; (Z \to ((Y \to Z) \to X))) \\ &\to\; (((X \to Y) \to Z) \to ((X \to Y) \\ &\to\; ((Y \to Z) \to X))) \end{aligned} \tag{1}$$

Substitution Assistant automatically generates new logical expressions when a user selects an arbitrary axiom and inputs arbitrary logical expressions as the substitution terms for that axiom.

Figure 1 shows a substitution flow on LASP.

Axioms are managed using a table as shown at the top of Fig. 1. When a user clicks on the axiom in the table which he wants to use, the Substitute Panel opens. If he inputs logical expressions for every propositional variable in Substitute Panel and clicks the OK button, then the result expression is added at the end of the proof list.

Such a feature has already been built into many interactive theorem proof systems. However, these systems are not easy to use in undergraduate classes.

3.3 Support for First-order Logic

LASP supports not only propositional logic but also first-order logic.

We have to consider whether the variable is free or bound when we want to perform substitution for a variable of an expression in first-order logic. For example, variable y of a logical expression (2) is bound by $\exists y$. Therefore, this variable cannot be substituted.

$$\forall x \exists y f(y, z) \tag{2}$$

Furthermore, any term including variable x or y cannot be also substituted into variable z of expression (2), because such substitution resulted bound by the quantifier. If expression $g(x)$ was substituted into z, the result would be the logical expression (3).

$$\forall x \exists y f(y, g(x)) \tag{3}$$

In the expression, variable x is bound by $\forall x$ and the semantic of the expression (3) is different from the expression (2). LASP generates an exception if an illegal substitution like the above occurs by using a substitution inhibition list.

3.4 Inference Assistant

Inference Assistant is implemented to reduce the practice time by reducing the time for operations such as copy and paste that take considerable time when performing by hand.

When a user selects an inference rule that he wants to apply and proven logical expressions that conform with inference rule, a new logical expression that corresponds to the rule is generated. The reason for the new logical expression is also generated, which reduces the amount of time needed to handwrite reasons.

Figure 2 shows the flow of the application of an inference rule on LASP. Inference rules and proven theorems are displayed in a table. First, a user clicks to select the inference rule which he wants to apply from the inference rule table. Second, he clicks to select the proven theorems matching the inference rule. Finally, he clicks the inference button. If he selects the proven theorems correctly, then a new logical expression is added at the end of the proven theorem table.

Figure 3 shows the screen-shot of LASP.

3.5 Deduction Theorem Assistant

The deduction theorem is a useful theorem for proving theorems efficiently. Therefore, LASP also supports the deduction theorem, which is the following.

From the fact that $\Gamma, P \vdash Q$, we obtain $\Gamma \vdash P \to Q$.

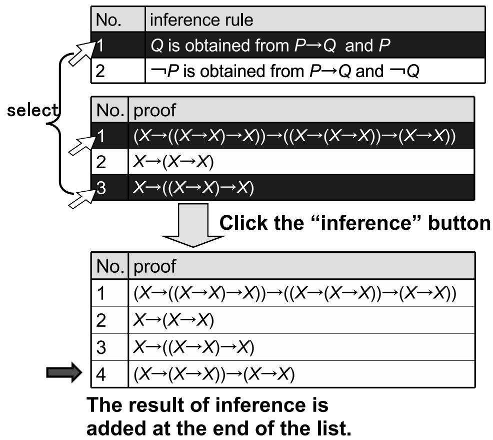

Figure 2: Flow of constructing a proof with LASP

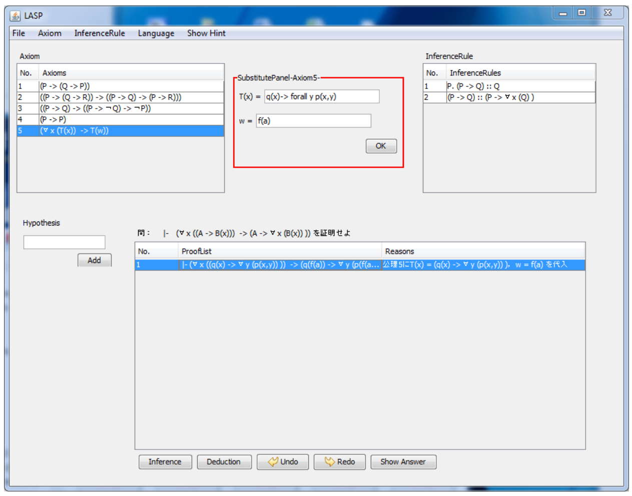

Figure 3: Screenshot of LASP

Table 2: Fill-in-the-blank hints

1		Hypothesis
2		“?” is substituted in axiom “?”
3		“A” and “?” are substituted for “P” and “Q,” respectively, in axiom “1”
4	$(\neg A \rightarrow A)$	Applying the inference rule “?”
5		“?” and “?” are substituted for “?” and “?” in axiom “?”
6	$((\neg A \rightarrow \neg A) \rightarrow \neg\neg A)$	Applying the inference rule “?”
7		Applying the inference rule “?”
8	$(A \rightarrow \neg\neg A)$	Deducing “?” from proven theorem “?”

Table 3: Milestone hints

1		
2		
3		
4	$(\neg A \rightarrow> A)$	Applying the inference rule “1” to proven theorems “1” and “3”
5		
6	$((\neg A \rightarrow \neg A) \rightarrow \neg\neg A)$	Applying the inference rule “1” to proven theorems “4” and “5”
7	$\neg\neg A$	Applying the inference rule “1” to proven theorems “2” and “6”
8		

First, a user selects a proven theorem to which he wants to apply the deduction theorem. Second, he clicks the deduction button. Then, the deduction panel opens. He clicks a radio button to select a theorem to which to apply the deduction theorem and then clicks the OK button. Finally, the result of applying the deduction theorem is added at the end of the proven theorem table.

3.6 Hint Features

The hint features provide three levels of hints for users who are unfamiliar with formal proofs. These features were implemented after obtaining feedback from Experiment 1, which is described in Section 4. The specific feedback is that it is difficult to solve exercises which have many steps by oneself. The type of hints prepared by LASP are as follows.

1. fill-in-the-blank hint
2. milestone hint
3. next step hint

Table 2 shows examples of fill-in-the-blank hints for a proof of $A \rightarrow \neg\neg A$ under the assumption that $P \rightarrow P$ has already been proved. This hint is generated randomly when LASP reads the data of an exercise.

Table 3 shows examples of milestone hints for the same proof. The hint feature shows expressions that can be proved by applying the inference rule.

We assume that a theorem $P \rightarrow P$ is given as axiom A4.

Table 4 shows a typical solution.

3.7 Undo/Redo Features

LASP supports undo/redo features. Learners can revise their proofs using these features.

It would be better if LASP supported a feature in which a user can edit his/her proof after he/she has completed it. However, such a feature might destroy the reasoning chain of the proof. Thus, the current version of LASP only supports Undo/Redo features. Undo/Redo works for as many steps as the length of the proof users have made.

3.8 Input/Output Features

Logical expressions are often long and deeply nested. Therefore, the string representation of a logical expression is sometimes difficult to understand. This feature visualises the parse tree of a logical expression (see the example in Fig. 4). Clicking a node, the subtree is expanded or collapsed (Fig. 5).

Table 4: Solution

1	A	Hypothesis
2	$\neg A \rightarrow \neg A$	P is substituted into axiom A4
3	$A \rightarrow (\neg A \rightarrow A)$	“A” and $\neg A$ are substituted for “P” and “Q,” respectively, in axiom “1”
4	$(\neg A \rightarrow A)$	Applying the inference rule “1” to proven theorems “1” and “3”
5	$((\neg A \rightarrow A) \rightarrow (\neg A \rightarrow \neg A) \rightarrow \neg\neg A)$	“$\neg A$” and A are substituted for “P” and “Q,” respectively, in axiom “3”
6	$((\neg A \rightarrow \neg A) \rightarrow \neg\neg A)$	Applying the inference rule “1” to proven theorems “4” and “5”
7	$\neg\neg A$	Applying the inference rule “1” to proven theorems “2” and “6”
8	$A \rightarrow \neg\neg A$	Applying the deduction theorem

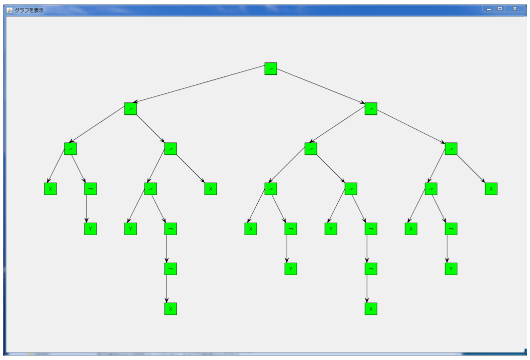

Figure 4: Parse tree representation

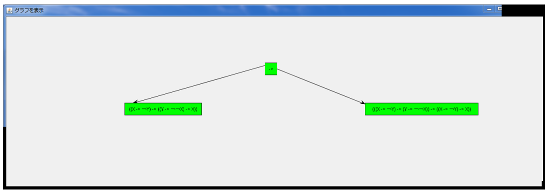

Figure 5: Parse tree representation (collapsed)

LASP also supports the LaTeX format for the output of constructed proofs. LaTeX supportof LASP also helps users, especially for producing reports or homework.

The exercises of LASP are input as XML files (Fig 6). Therefore, a teacher can easily prepare new sets of exercises with hints.

4 Evaluation

This section describes an evaluation of LASP. We performed two experiments. Experiment 1 was conducted in a mathematical logic class. The subjects were 50 undergraduate students. Experiment 2 was conducted after reflecting on the feedback received from Experiment 1. In this experiment, the number of subjects was 16. The subjects in Experiment 2 were a doctoral student, ten master course students, and four undergraduate students.

4.1 Goals of the Evaluation

The objective in these experiments was as follows: to measure the degree of effectiveness for users, and to collect feedback to make LASP more useful.


```
<?xml version="1.0" encoding="Shift_JIS" ?>
- <root>
  - <axioms>
      <axiom>P imply (Q imply P)</axiom>
      <axiom>(P imply (Q imply R) ) imply ((P imply Q) imply (P imply R))</axiom>
      <axiom>(P imply Q) imply ((P imply not Q) imply not P)</axiom>
    </axioms>
  - <inference_rules>
      <inference_rule>P . P imply Q :: Q</inference_rule>
    </inference_rules>
  - <proof>
      <question>not Y</question>
      <hypothesis>not (X -> Y)</hypothesis>
      <hypothesis>not X</hypothesis>
    </proof>
</root>
```


Figure 6: The output XML file

4.2 Items of the Evaluation

In Experiment 1, we mainly investigated how effectively a user can solve an exercise and the usability of LASP. The degree of efficiency was measured as the time it took subjects to finish solving the given problems.

As the evaluation of usability, we prepared questionnaires to research how subjects can become familiar with LASP and what kind of interface is needed to enhance usability. The items of the questionnaires are summarised as follows.

$Q1$ The degree of user-friendliness of Substitute Assist.

$Q2$ The degree of user-friendliness of Inference Assist.

$Q3$ The degree of user-friendliness of LASP as a whole.

$Q4$ The degree of efficiency improvement using LASP.

$Q5$ The degree of reduction of trouble in proving from using LASP.

$Q6$ The degree of effects of learning formal proof.

The items are on a scale of one to five, where a five means a high degree. A free comment space to collect opinions about LASP or the experiment is also included.

4.3 Procedure of Experiments

The procedure of Experiment 1 is as follows.

1. We divide all subjects into six groups.
2. Subjects solve two practice problem by hand.
3. Two weeks later, subjects solve two practice problems by using LASP.
   (a) We distribute tool manuals to all subjects.
   (b) We let subjects use LASP to solve sample exercises for 20 minutes.
   (c) Subjects solve the exercises.
4. Subjects answer the questionnaire.

There are four exercises in all. The order of the exercises is randomly chosen for each group. The time limit that a subject has to solve an exercise is 15 minutes.

The procedure of Experiment 2 is as follows.

1. We distribute tool manuals and textbooks about mathematical logic to all subjects.
2. We give all subjects one hour to solve four sample exercises and familiarise them with LASP.
3. Subjects alternate between solving exercises by hand and using LASP.
4. Subjects answer the questionnaires.

Table 5: Results of the questionnaires in Experiments 1 and 2

item	Exp.1	Exp.2
$Q1$	3.70	4.69
$Q2$	3.28	4.56
$Q3$	2.84	4.13
$Q4$	2.88	3.63
$Q5$	3.28	3.75
$Q6$	2.84	3.63

Table 6: Problems in Experiment 1

	problem
$Q1$	$A \to \neg\neg A$
$Q2$	$A \to B \vdash \neg B \to \neg A$
$Q4$	$\forall x(P(x) \to Q(x)) \to (\forall x P(x) \to \forall x Q(x))$

The time for subjects to familiarise themselves with LASP is one hour in Experiment 2 because twenty minutes was not enough in Experiment 1. There are four exercises in all, the same as in Experiment 1. The order of exercises differs by group. The way of measuring the solving time is the same as in Experiment 1.

4.4 Results of the Experiments

4.4.1 Results of the Questionnaires

Table 5 shows the average scores of the questionnaire items for Experiments 1 and 2.

The following feedback was obtained from the free comment space of Experiment 1.

- It would be better if the nested logical expressions were shown clearly by using a colour-coded fonts.
- It is troublesome that users have to use a mouse.
- Users want to adjust the window size freely.
- Panel alignment should be improved.
- Shortcut keys should be provided.
- A hint feature should be provided.
- It should show the answer.
- Users seem to be able to become familiarised with it, given enough time.

The following feedback was obtained from the free comment space of Experiment 2.

- LASP should notify the user clearly when the proof is correctly constructed.
- It is troublesome for users to write expressions including quantifiers because users have to input "forall" to display ∀.
- Users are not allowed to delete logical expressions except the latest one.
- Users are not allowed to add logical expressions except at the end of their list.
- It would be better if there was a memo panel for planning proof tactics.
- It would be better if there were a feature for users to make their own exercises.

Table 7: Number of subjects who correctly solved an exercise in Experiment 1 (/total number of subjects)

	handwriting	LASP
$Q1$	5/23	7/23
$Q2$	6/24	7/23
$Q4$	0/27	1/22

Table 8: Average solving time in Experiment 1

	handwriting	LASP
$Q1$	9m13s	10m04s
$Q2$	6m56s	8m15s
$Q4$	No one solves	8m18s

- Shortcut keys should be implemented.
- Panel alignment should be improved.
- It is hard to see the nested expressions.
- Hints should be improved.

From the comments, we conclude that the following are the advantages of LASP relative to doing proofs by hand.

- Users can reduce their amount of effort.
- Users can reduce the number of careless mistakes they make.
- It is easier to solve exercises because users can try many tactics within a short time.
- Amount of time which users can use to learn by making mistakes is increased because less time and effort is required to make substitutions and do inference.

4.4.2 Results for Solving Time

Table 6 shows the problems used in Experiment 1. Every problem assumes that theorem $P \to P$ has already been proved.

Tables 7 and 8 show the results of Experiment 1. We omit the results of Exercise 3 because it involves a mistake. Table 7 shows the number of subjects who solve exercises correctly (S_c) and all subjects (S_{all}). Table 8 shows the average solving time of subjects who correctly solved the exercise.

Table 9 shows the problems used in Experiment 2. Every problem assumes that the theorem $P \to P$ has already been proved.

Tables 10 and 11 show the results of Experiment 2. Table 10 shows S_c/S_{all} and Table 11 shows the average solving time for subjects who correctly solved the exercise.

5 Discussion

First, we consider the usability which is evaluated in Experiment 1. We can see that users are unsatisfied with LASP's usability from the results of $Q1, Q2$, and $Q3$ in Experiment 1, shown in Table 5. This is also shown by the results for $Q6$ in Experiment 2. Therefore, LASP's user interface should be improved. In order to improve the interface, we have to consider the free comments. We think that implementing shortcut

Table 9: Problems in Experiment 2

	problem
$Q1$	$\neg A \to B \vdash \neg B \to A$
$Q2$	$\neg X \to (X \to \neg Y)$
$Q3$	$(\neg Y \to \neg X) \to ((\neg Y \to X) \to Y)$
$Q4$	$\forall x(A \to B(x)) \to (A \to \forall x B(x))$

Table 10: Number of subjects who correctly solved an exercise in Experiment 2 (/total number of subjects)

	handwriting	LASP
$Q1$	5/8	2/8
$Q2$	4/8	4/8
$Q3$	4/8	2/8
$Q4$	2/8	2/8

keys and a feature which shows the users that they have finished correctly proving a proof are easy tasks.

Second, we consider the efficiency of proving, which was mainly evaluated by Experiment 2. From the results of $Q1, Q2$, and $Q3$ shown in Table 5, we can see that the substitution assistant feature and the inference assistant feature have contributed sufficiently to the efficiency of proving. Users also feel that. In addition, twelve out of sixteen subjects in Experiment 2 comment that LASP can reduce the required effort relative to doing problems by hand. Therefore, the goal of LASP's development seems to have been sufficiently achieved. The reason that the average score for $Q3$ is less than those of $Q1$ and $Q2$ seems to come from dissatisfaction with LASP's usability. Therefore, we should implement shortcut keys and other features.

From the results of Tables 8 and 11, we can see that whereas proving by LASP is slower than by hand in Experiment 1, the opposite result was obtained in Experiment 2. The reason for this seems to be that, while the time during which subjects could familiarise themselves with LASP was twenty minutes in Experiment 1, it was an hour in Experiment 2. Therefore, we conclude that if users are familiar with LASP, then they can solve practice problems more effectively.

The sample size of Experiment 2 was insufficient; we should perform experiments on a larger scale to prove the correctness of this hypothesis.

From the results shown in Tables 7 and 10, we can see that the number of subjects who can correctly solve exercises using LASP is the same as the number who can correctly solve exercises by hand. Thus the results suggest that the increased efficiency may not influence the correctness of answering questions. We cannot simply say that LASP helps students to easily solve exercises on logic, because fewer subjects could solve exercises Q1 and Q3 using LASP than by hand. However, in the handwritten answers, there were a mistake in applying inference rules and also two substitution mistakes. Using LASP, users can avoid careless mistakes because it shows an error message when users wrongly apply inference rules.

Table 11: Average solving time Experiment 2

	handwriting	LASP
$Q1$	10m35s	5m57s
$Q2$	7m19s	6m04s
$Q3$	8m48s	5m54s
$Q4$	12m56s	5m19s

In general, a learner needs trial and error to obtain a correct answer. LASP can store every step (trial) of a proof that the learner produces. From such a set of steps of a proof, the learner can select the essential sequence of the proof. For a simple proof task, the learner easily finds the essential sequence of the proof; however, it is difficult for users in the case of a complex proof task. Therefore, a feature giving assistance in the rewriting of the proof is being considered. Such a feature can be implemented by an automatic choice of directly relating steps of a focusing step. We want to add such a feature to LASP.

In conclusion, LASP is effective for learning formal proofs.

6 Conclusion

This paper describes our learning assistant system for formal proof, LASP. LASP has features such as inputting exercise data, a substitution assistant, an inference rule assistant, and so on. From two experiments, we obtained results implying that LASP is effective, although the user interface should be improved. Future work includes the improvement of the user interface and actual application in a class.

We strongly believe that the hint feature is useful for a learner to reach the correct answer. However, the design of hints requires as much care as the design of exercises because too much information can let a learner obtain the correct answer too easily; on the other hand, a little information seldom helps a learner obtain the correct answer. Therefore, the teacher has to construct hints manually. Of course, the system can help the teacher to design hints. To develop such an assistance feature for designing hints and to evaluate the power of the hint feature is part of our future work.

Acknowledgments

This work is being conducted as a part of Stage Project, the Development of Next Generation IT Infrastructure, supported by the Ministry of Education, Culture, Sports, Science and Technology, Japan, as well as a Grant-in-Aid for Scientific Research C (21500036). We also thank all subjects who participated in the experiments.

REFERENCES

[1] D. Abraham, L. Crawford and et al, "A tool to practice formal proofs," Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp.7–8 (2001).

[2] C. Baier and J. Katoen, "Principles of model checking," MIT Press (2008).

[3] D. Barker-Plummer, J. Barwise, and J. Etchemendy in collaboration with A. Liu, "Tarski's World: Revised

and Expanded," Center for the Study of Language and Inf (2007).

[4] J. Barwise and J. Etchemendy, "Turing's World 3.0: An Introduction to Computability Theory," University of Chicago Press (1993).

[5] J. Barwise and J. Etchemendy, "Computers, visualization, and the nature of reasoning," The digital phoenix: How computers are changing philosophy, pp.93–116 (1998).

[6] Y. Bertot and P. Castéran, "Interactive theorem proving and program development: Coq'Art: the calculus of inductive constructions," Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag (2004).

[7] A. Bove, P. Dybjer and U. Norell, "A brief overview of agda — a functional language with dependent typess," In Proceedings of the 22nd International Conference on Theorem Proving in Higher Order Logics (TPHOLs '09), pp.73–78, (2009).

[8] B. Buchberger, A. Crăciun and et al, "Theorema: Towards computer-aided mathematical theory exploration," Journal of Applied Logic, Vol.4, No.4, pp.470–504 (2006).

[9] R. Dyckho, "MacLogic: A Proof Assistant for First Order Logic on the Macintosh," Computational Science Division, University of St. Andrews (1989).

[10] D. Gillet, A. Vu Nguyen Ngoc and Y. Rekik, "Collaborative web-based experimentation in flexible engineering education," IEEE Transactions on Education, Vol.48, No.4, pp.696–704 (2005).

[11] P. Henderson, "Mathematical reasoning in software engineering education," Communications of the ACM, Vol.46, No.9, pp.45–50 (2003).

[12] H. Kaufmann and D. Schmalstieg, "Mathematics and geometry education with collaborative augmented reality," Journal of Computer and Graphics, Vol.27, No.3, pp.339–345 (2007).

[13] K. Miyazawa, "LASP" (2010). `http://sdl.ist.osaka-u.ac.jp/~k-miyazw/`.

[14] C. Shubert, I. Ceraj and J. Riley, "Bringing research tools into the classroom," Journal of Computers in Mathematics and Science Teaching, Vol.28, No.4, pp.405–421 (2009).

(Received February 27, 2012)
(Revised March 1, 2013)

Kiyoyuki Miyazama received the BE and MI degrees in computer science from Osaka University in 2009 and 2011, respectively. His research interests include model driven architecture and education of formal approach.

Kozo Okano received the BE, ME, and PhD degrees in information and computer sciences from Osaka University in 1990, 1992, and 1995, respectively. Since 2002 he has been an associate professor in the Graduate School of Information Science and Technology, Osaka University. In 2002, he was a visiting researcher of the Department of Computer Science, University of Kent at Canterbury. In 2003, he was a visiting lecturer at the School of Computer Science, University of Birmingham. His current research interests include formal methods for software and information system design. He is a member of IEEE, IEICE of Japan, and IPS of Japan.

Shinji Kusumoto received the BE, ME, and DE degrees in information and computer sciences from Osaka University in 1988, 1990, and 1993, respectively. He is currently a professor in the Graduate School of Information Science and Technology at Osaka University. His research interests include software metrics and software quality assurance techniques. He is a member of the IEEE, the IEEE Computer Society, IPSJ, IEICE, and JFPUG.

Experimental transformations between Business Process and SOA models

Akira Tanaka[†], and Osamu Takahashi[‡]

[†]view5 LLC, Japan
[‡]School of Systems Information Science, Future University Hakodate, Japan
a.tanaka@view5.co.jp, osamu@fun.ac.jp

Abstract - in designing enterprise IT systems, two major architectural styles exist today: process-oriented and service-oriented architectures. Either one of them can be used to define behavioral aspects of the business specifications. In reality, a process can make use of various services, and a business service can be implemented as a process. This duality applies to such technology as BPMN and SOA. In RM-ODP standard, however, both are part of a standard viewpoint language, and they complement with each other. In this paper, using a textual domain specific language and a tool supporting it to capture the essence of those modeling languages, we examine the relationship between process-based specifications and service-based specifications for a typical buy-sell-ship business process or collaboration. Architectural comparison is done by examining a model transformation of process to service, and service to process. The difference of the two types of model and the implication of the results are discussed.

Keywords: business process; service-oriented architecture; enterprise architecture; RM-ODP; model transformation.

1 INTRODUCTION

Enterprise architecture is widely used as a way to describe overall architecture of enterprise systems. There are many approaches to define enterprise architecture. For instance, to describe everything from hardware components to business strategies, a matrix of concerns and perspectives is used in Zachman Framework[1], or a set of customized architectures such as application architecture to capture different aspects is used in Federal Enterprise Architecture[2] initiatives. There are also requirements to harmonize businesses and ITs. In this context, the importance of business specifications, which describes “what to achieve” at business level, is being recognized.

There are a variety of approaches to define business specifications: business process oriented approach that uses “Business Process Model and Notation” (or BPMN[3]), business rule oriented approach that uses “Semantics of Business Vocabulary and Business Rules” or (SBVR[4]), business events based approach such as “Event Driven Architecture” (or EDA) and “Complex Event Processing” (or CEP[5]), business service oriented approach such as “Service Oriented Architecture” (or SOA[10]), and many more. Among them, two major architecture styles exist, which are process-oriented and service-oriented architectures. They are mainly used to define behavioral aspects of the business specifications.

According to BPMN specification, “business process is a defined set of business activities that represent the steps required to achieve a business objective.” This implies, when top down design approaches are taken, a business objective is set first, and a process or a set of processes is/are defined to achieve the objective. The IT system will be designed to provide necessary functionalities to the defined steps in the process. In this approach, interested behaviors or interactions exist between steps and among all the participants.

According to SOA Reference Model, “A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.” In this approach, most application elements will be implemented as services so that client software can find and consume necessary services to achieve its goal. The interactions in this model are between consumer and provider of the service. Orchestration of the services is not within the scope of this architecture.

Both styles are used as foundational architectures when developing enterprise IT systems. It should be noted, however, that a process can be decomposed into steps, each of which may consume services. And, a service can be implemented as a process. This duality applies to business systems designed based on current technologies such as “Unified Modeling Language”[1] (or UML[6]), BPMN and SOA.

The issue we have is “are they really different, or are they different sides of the same coin?” In other words, are they only different in architecture styles and equivalent in capabilities? And if so, how can we measure the equivalence? In this paper, we will examine this issue using modeling technique including UML, “Domain Specific Language” (or DSL[8, 9]), and model transformations. If they are essentially equivalent, there should be correspondences between them, that is, a service based model should be able to be transformed into a process based model, and a process based model should be able to be transformed into a service based model.

We will first look at business process oriented approach and examine how much it can be mapped to SOA approach. We will then look at service based approach and examine how much it can be mapped to process oriented models. In doing this, we will use DSL and model to text transformation tool.

[1] UML is a standard graphical modeling language for analyzing, designing and implementing software-based systems. The current version is UML 2.0.

2 BUSINESS PROCESSES

Business specifications are usually the most examined specification in Enterprise Architectures, since most business users, in addition to technology providers, would need to review it to see if it correctly captures the business requirements. When top down approach is taken, it usually starts with analysis of business environment, establish a new goal and strategies, design business processes to achieve the goals. There are multiple choices in diagramming business processes. UML Activity Diagram and Business Process Modeling and Notations (BPMN) are the most used ones. In this paper, we use ODP ([7] see IV) Process diagrams that is a slight extension to UML Activity Diagram, since it has necessary characteristics to do the experiment.

The diagram on the right side of this page (Figure 1) shows a sample purchasing business process among three parties, buyer, seller, and shipper in ODP Process Diagram. Each lane represents role and behavior of the party, and each step is represented as an Action internal or external to the lane. In case of external Action, the control flow crosses the lane with or without artifact passing. Artifact, represented as ObjectNode, is there to capture necessary business information to be passed. There are split/merge used to control, i.e. to create and conclude, parallel activities. This is almost the basic Activity Diagram except for applied ODP stereotypes. The dotted lines are there to show logical grouping of the steps that have certain meaning in the application, e.g. placing an order. Note that these dotted lines are not part of standard notation, and they should be read as additional comment. Although it is possible to group steps using sub-process, it does not provide improvement in readability of the process, and therefore we did not take that approach.

In summary, this process diagram shows participants of the purchasing process and a collection of necessary steps in a prescribed manner leading to the objective. IT systems will be designed to support some portions of the steps. This style is effective when an IT system is to be built against pre-defined business processes (i.e. what needs to be done in what order).

3 SERVICE OR BUSINESS SERVICE

The definition of term "Service" in SOA is still under open discussion. However, there is one in OASIS's SOA Reference Model[10], which is "A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description." OMG's SoaML[11], which is a UML Profile for Service oriented architecture Modeling Language, is a standard to describe SOA based models. A slight modification of above definition was used there, which says "A service is value delivered to another through a well-defined interface and available to a community (which may be the general public). A service results in work provided to one by another." In this UML Profile, various SOA concepts are defined. For instance, a community is defined as a place for participants consume/provide services to each other. For each pair of participants they have service contract that govern the behavior aspects when consuming/providing services. As for diagramming, it mainly uses UML Collaboration Diagram, Class Diagram and Component Diagram.

If compared with the previous process model, participant can be considered as role. The diagram on the right side of this page (Figure 2) shows a sample high level Service Architecture using SoaML to represent Buy-Sell-Ship collaboration.

Note that there are other kinds of diagrams associated with this high level description. For instance, Service Contract Diagram contains two roles, consumer and provider, with a sequence diagram specifying service message exchanges when provided services are consumed.

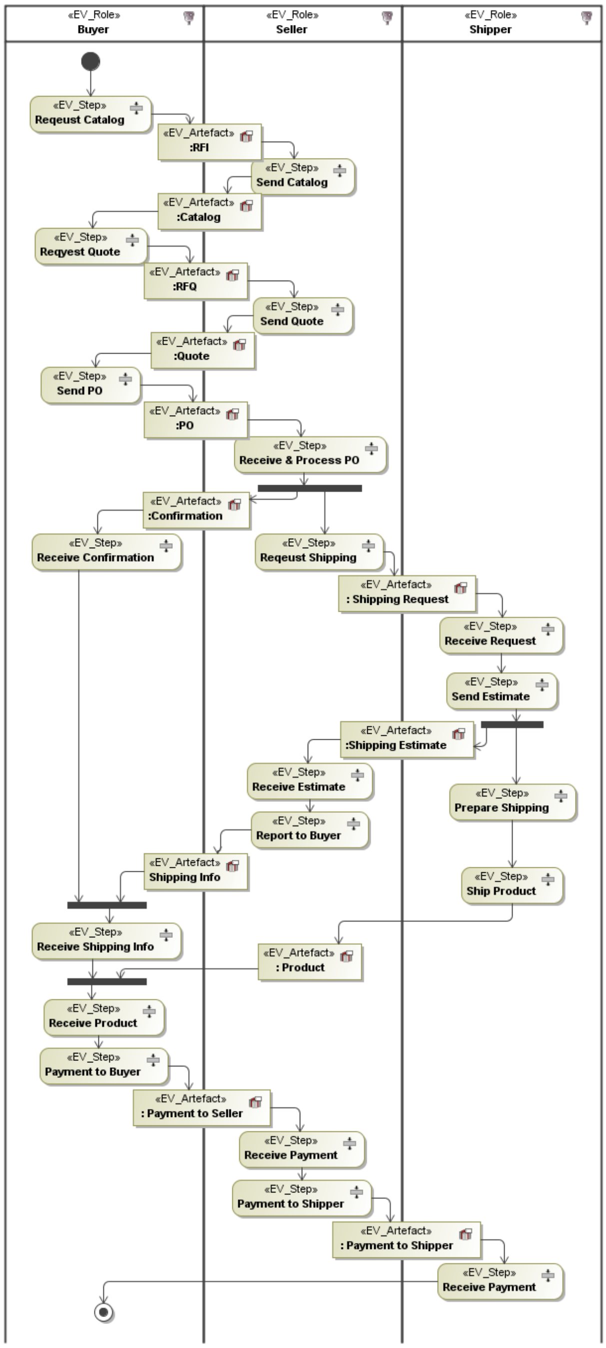

Figure 1: Sample ODP Purchasing Business Process

In summary, the service diagram (Figure 2) shows participants of the collaboration and service contracts, which includes interface and behavior definition, between service consumers and service providers. Note that there is no “steps taking place in a prescribed manner” defined. IT systems will be designed to support service providers and consumers. This style is effective when an IT system is to be built using existing or to be built services, such as newly developed internal services, wrapped legacy functions, or external services using web services, with a flexibility of clients’ choice of the services.

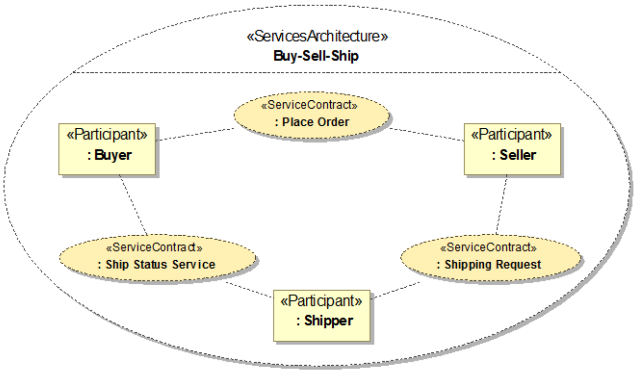

Figure 2: Sample Service Architecture

4 RM-ODP

RM-ODP stands for Reference Model for Open Distributed Processing, which is a family of international standards for developing standards for open distributed processing systems. It is a set of reference models, and it also has UML Profile standard to represent its concepts using UML tools. This standard is used as an open Enterprise Architecture, and we use this standard with associated UML Profile (e.g. in Fig. 1) to show something not biased to specific process modeling notations. It defines five standard viewpoints, but we will use or refer to only three of them in this paper: Enterprise, Information, and Computational viewpoints.

5 DOMAIN SPECIFIC LANGUAGE (DSL)

According to Domain-Specific Languages[8], domain –specific language is defined as “a computer programming language of limited expressiveness focused on a particular domain.” DSL could be graphical or textual, could be internal (designed based on general purpose language) or external (having no specific host language). In this paper, textual and external DSLs for process oriented modeling (ProcessDSL) and service oriented modeling (ServiceDSL) are described and used.

6 PROCESS TO SERVICE MODEL TRANSFORMATION

Model transformation achieves one source model described in one specific language to be converted into target model described in other specific language, without violating rules for those languages. Typical example is to transform UML Class Diagram to Relational Table. Four layer meta architecture is usually used to explain the mechanism. As a standard, OMG’s MOF/QVT[12] is the best known one. As open source projects, widely known examples are eclipse ATL[13, 14] and QVT.

The next table shows metamodel and main elements of Process models and Service models related to our sample process.

Table 1: Metamodel Elements

	Business process	Service
Metamodel	Process related part of RM-ODP Enterprise Viewpoint metamodel	SoaML metamodel
Main Elements	Object, Role, Process, Step, Action, Activity, Artefact, Interaction	Participant, Service Contract, Service Architecture, Service Interface, Service Choreography, Message, UML Collaboration/Component

The following is a summary of how service metamodel element could be created with given process metamodel elements.

A. Participant/UML Component

A Participant (actually a Type) is equivalent to a Role in process. Since Participant deals with computation, a computational object with the same name should be introduced or assumed in the Process model side.

B. Service Contract

Service Contract can be considered as a concept representing interaction or behavior between two roles. ODP’s Interaction is the closest concept, but it is not really a part of process modeling. Service Contract uses Sequence Diagram to represent the behavioral aspects.

C. Service Architecture/UML Collaboration

Service Architecture’s closest concept is a collection of Interactions among all the involved roles.

D. Service Interface

The concepts like Interface, Operation, and Signature belong to Computational Object in ODP. If we could assume the existence of Computational Object with the same name as Participant, they are the corresponding elements.

E. Service Choreography

Service Choreography defines ordering of service messages between service consumer and provider. This can correspond to an ordered sequence of ObjectFlow involving the corresponding two roles in the process model.

F. Message

Information viewpoint of RM-ODP is the viewpoint where all the concerns on information within the system are defined. However, in Enterprise and Computational viewpoint, there is a need for information model and they are created based on the one defined in Information viewpoint. The Message data types are a collection of data types and structure of data elements visible in Enterprise viewpoint, and therefore those should be prepared as a part of process model.

From above, we can observe that when transforming a process to a service, the process needs to be decomposed into an ordered set of two party interactions, and interaction should be brought into a process model.

Figure 3 is a sample ODP Interaction Diagram showing Buyer as an initiator of the interaction, Seller as a responder of the interaction, and the references to various artifacts that are actually electronic document. The behavior of both sides is described using UML State Machine, which is different from the case with Service Contract.

When considering the model transformation, this Interaction Diagram could act as an intermediate from process to service, meaning that we can transform a process model to interaction model, and then transform it to service model.

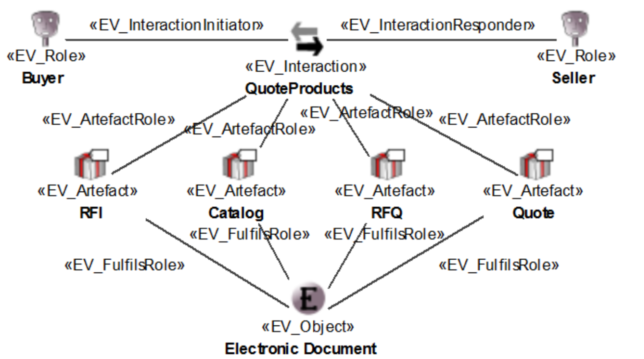

Figure 3: ODP Interaction Diagram

7 SERVICE TO PROCESS MODEL TRANSFORMATION

This section considers the reverse of the previous section, which is about transformation from service model to process model. Service model here is a model based on SoaML.

A. Object

SoaML's instance of Participant can be considered as Enterprise Object in a limited sense, since it also consumes and provides services like Computational Object. Message is a good source for defining Information Object.

B. Role

Participant is almost the equivalent to role. At the same time, Participant can be understood as Computational Object, which takes all the interface information from the Participant.

C. Process/Activity

Service Choreography specifies dynamic binary relationship of the behavior, and can be used to construct a portion of a Process. However, constructing whole process is not possible, since there is no orchestration information is available for construction of the whole process, including where to start the process.

D. Step/Action

Step/Action means action execution, which is defined in Service Contract and Service Choreography. However, it is not possible to generate Steps/Actions which do not correspond to interactions such as internal actions.

E. Artefact

Message is the only element to map to Artefact.

F. Interaction

Service Contract corresponds to Interaction where Consumer side corresponds to Initiator side and Provider side corresponds to Responder side.

8 MODEL TRANSFORMATION METHOD USING TEXUAL DSLS

When we refer to "model transformation," it usually means transforming models created with UML tools or some other specific tool such as BPMN tools (graphical tools). Those models are actually saved as text file, for instance as a form of XMI[15] or XML, and then transformation logics are applied to it. However, UML itself is a complex specification and that is reflected in XMI. A simpler way to experiment some modeling issues without involving too much complexity was needed, and that was the reason we took DSL.

We will now explain a method of model transformation using textual DSL. With textual DSL such as the one developed with eclipse/Xtext[16], a grammar is first defined, and the grammar based model editor is generated so that user can create his/her own model based on the DSL.

Suppose you have two textual DSLs: ProcessDSL for process modeling and ServiceDSL for service modeling. If you define a grammar for ProcessDSL, you get the ProcessDSL editor. The same is true for ServiceDSL. Once a process model is defined, a template is applied to the model to generate output text (e.g. source code or XML file). Here, it is possible to design a template to generate text, which has a structure that ServiceDSL editor imposes. This is not always possible, since the source model may not contain necessary information to transform to. But, if it did, this model to text transformation works as a model to model transformation. This is the basic idea we used for Process to Service and Service to Process model transformations.

In order to achieve this, we have created above DSLs as simple textual DSLs to capture core concepts of ODP Process Diagram (or UML Activity Diagram) and SoaML Diagram respectively. They are simple, because not all concepts are used and some complex concepts were simplified to some extent in the grammar.

The tool used is eclipse/Xtext and its integrated model to text transformation engine Xpand/Xtend. But, with any other textual DSL tooling, such as Spoofax[17] or MPS[18], this can be done in the similar way.

Figure 4 shows a portion of the Process DSL grammar.

Using generated DSL editor, a process model in this ProcessDSL is created (Figure 5), which is done by typing, not by applying model to text transformation from the process model (Figure 1). The last step is to define a template to generate text, which is explained in the next page. The Xtext grammar files and sample model are published on the following web site[19].

We also created InteractionDSL based on ODP concept of Interaction, and a sample model is shown in Fig. 6. With this InteractionDSL, multiple artifacts are allowed to be shown, but in our transformation rule, only one artifact per interaction is generated. This is because ODP interaction diagram shows aggregation of multiple interactions, but our process to service transformation is targeted to each crossing the lane interaction only.

```
Model:
    processes+=Process*;
Process:
    'process' name=ID '{'
        lanes+=Lane*
    '}' ;
Lane:
    'role' name=ID '{'
        processFlows+=ProcessFlow*
    '}' ;
ProcessFlow:
    InitialNode | EndNode | Step | Artifact | ParallelSplit |
    ParallelMerge | ConditionalSplit | ConditionalMerge ;
InitialNode: 'start' name=FQN ;
EndNode: 'end' name=FQN;
Step: 'step' name=FQN ('{'
        ('description' description=STRING)?
        fromNode=FromNode
        toNode=ToNode
    '}')? ;
Artifact: 'artifact' name=FQN ('{'
        ('description' description=STRING)?
        fromNode=FromNode
        toNode=ToNode
    '}')? ;
ParallelSplit: 'parallel' 'split' name=FQN ('{'
ParallelMerge: 'parallel' 'merge' name=FQN ('{'
ConditionalSplit: 'conditional' 'split' name=FQN ('{'
ConditionalMerge: 'conditional' 'merge' name=FQN ('{'
FromNode:
ToNode:

FQN: ID('.'ID)* ;
```

Figure 4: Partial ProceeDSL grammar

```
process Buy_Sell_Ship {
    role buyer {
        start StartOfPurchasingProcess
        step RequestForInformation {
            description "Gathering Product Information"
            from StartOfPurchasingProcess
            to seller.ReceiveRFI with Artifacts.RFI
        }
        step ReceiveCatalog {
        conditional split PurchaseFromCatalog {
        step SendQuoteRequest {
        step ReceiveQuote {
        step SendPO {
        step ReceivePOConfirmation {
        parallel merge MergeWithInformation {
        step ReceiveShipmentInformation {
        parallel merge MergeWithProductsDelivery {
        step ReceiveProducts {
        step PaymentToSeller {
        end EndOfPurchasingProcess
    }
    role seller {
    role shipper {
    role Artifacts {
        artifact RFI {
            description "Reqeust for information document"
            from buyer.RequestForInformation
            to seller.ReceiveRFI
        }
        artifact Catalog {
        artifact QuoteRequest {
        artifact Quote {
        artifact PO {
        artifact POConfirmation {
        artifact ShippingRequest {
        artifact ShippingCostEstimate {
        artifact ShipmentInformation {
        artifact Products {
        artifact SellerPayment {
        artifact ShipperPayment {
    }
}
```

Figure 5: Sample Process Model in ProcessDSL

```
interaction PlaceOrder {
    initiator Buyer
    responder Seller
    artifact RequestForInformation
    artifact Catalog
    artifact PurchaseOrder
    artifact PurchaseOrderConfirmation
    artifact Payment
}
```

Figure 6: Sample Interaction Model in InteractionDSL

Regarding tools for model transformation, widely used ones are eclipse/ATL and QVT. They could also be used to execute model transformations described in this paper.

The Xpand template used is shown in Fig. 7, which should be considered as a sample. It does the following: a) import metamodel, b) import helper functions, c) declare this as a transformation against Process, d) specify output file, e) define transformation processing for each lane, f) take only steps passing artifact to other lanes, g) generate Service Interfaces, h) generate Participants, and i) generate Service Contracts with model information.

```
«IMPORT jp::ac::fun::xtext::process::processDsl»
«EXTENSION templates::Extensions»
«REM» SoaML Service Architecture «ENDREM»
«DEFINE main FOR Process-»
«FILE this.name+".soa"-»
ServiceArchitecture «this.name» {
  «FOREACH this.lanes AS lane-»
    «FOREACH lane.processFlows.typeSelect(Step) AS node-»
    «IF node.fromNode.fnode.eContainer()==node.eContainer()-»
    «ELSE-»
      «IF node.fromNode.isArtifactProvidedFrom()-»
  ServiceInterface «node.fromNode.fnode.name» {
  }
      «ENDIF-»
    «ENDIF-»
    «ENDFOREACH-»
  «ENDFOREACH-»
  «FOREACH this.lanes AS lane»«IF lane.name=="Artifacts"»
    «ELSE-»
  Participant «lane.name» {
  }
    «ENDIF-»
  «ENDFOREACH-»
  «FOREACH this.lanes AS lane»
    «IF lane.name=="Artifacts"-»
      «FOREACH lane.processFlows AS msg-»
  Message «msg.name»
      «ENDFOREACH-»
    «ENDIF-»
  «ENDFOREACH-»
  «FOREACH this.lanes AS lane»
    «IF lane.name=="Artifacts"-»
      «FOREACH lane.processFlows AS msg-»
  Message «msg.name»
      «ENDFOREACH-»
    «ENDIF-»
  «ENDFOREACH-»
  «FOREACH this.lanes AS lane-»
    «FOREACH lane.processFlows.typeSelect(Step) AS node-»
    «IF node.toNode.tnode.eContainer()==node.eContainer()-»
    «ELSE-»
      «IF node.toNode.isArtifactProvided()-»
  ServiceContract «node.name» "" among {
    consumer «node.eContainer().getContainerName()»
      AsConsumer
      performed by «node.eContainer().getContainerName()»
    provider «node.toNode.tnode.eContainer().getContainerName()»
      AsProvider
      performed by «node.toNode.tnode.eContainer().getContainerName()»
    «node.eContainer().getContainerName()»AsConsumer
    ->«node.toNode.tnode.eContainer().getContainerName()»AsProvider
      «node.toNode.anode.name»
  }
      «ENDIF-»
    «ENDIF-»
    «ENDFOREACH-»
  «ENDFOREACH-»
}
«ENDFILE-»
«ENDDEFINE»
```

Figure 7: Sample model transformation definitions

As shown, if the source model contains enough information with formally defined grammar or metamodel, it is possible to create text file based on available information and by navigating the model elements. We have generated a small number of text files using this method, and will examine those in the next section.

One question may be asked about how to make sure that the DSL targeting only core concepts can be used in the research like this. It is different, but if it contains major elements with right relationships, it is possible to compare the grammar or generated EMF ecore file with that of full modeling language to see if there is any major inconsistencies or issues in using it in the research.

9 RESULTS OF PROCESS TO SERVICE TRANSFORMATION

Now let us examine what the transformation produced. The first is a result around Service Architecture.

A. Participant/Service Contact/Service Architecture

From the used template above, here is a summary of what operations were given to the sample process model.

1) *Service Architecture name is derived from Process's name.*

2) *Participant name is derived from lane's name.*

3) *Service Contract is defined for the step that passes Artefact across the lanes. Other types of actions will be discarded. Service Contract name is derived from the step name that initiates interaction.*

If we modify the grammar to include marking to show the logical boundary of the application, it may become possible to have coarse grained Service Contract with multiple Artefact, but that will introduce another requirement on the dependencies between Artefact.

The generated textual model is shown in Fig. 8. It is generated from sample process model in ProcessDSL (Figure 5) by applying the process to service model transformation (Figure 7). Note that the model data below has been imported into ServiceDSL editor.

B. Service Interface

Service Interface in SoaML is functional elements, and is more like interface and signatures in Computational Object. The best way is to define Computational Object with process definition, but that will lead to a different world. Based on the generated Service Contract, it is safe to assume that Participant on receiving side have capability to process the Artefact passed from the other side. This implies that there exists Service Interface on the Service Provider Component. However there is no information about signatures in process models in general, it would not be possible to generate Message elements either. Therefore the rule applied was Service Interface is derived from the node (name) that receives Artifact.

C. Service Choreography

Service Choreography is a set of defined sequence of service interactions between the two Participants, which is specified using UML Sequence Diagram. This contrasts with UML Activity Diagram we used to specify process. Although not included in Figure 8, it is possible to collect interactions between different lanes in the process diagram.

Although BPMN, Activity Diagram, and SoaML are all graphical modeling language, we applied our method to define simplified textual DSL, and were able to transform a process model to a service model, although in a limited manner.

```
ServiceArchitecture Buy_Sell_Ship {
    ServiceInterface SendCatalog {
    ServiceInterface SendQuoteResponse {
    ServiceInterface AcceptPO {
    ServiceInterface RequestForInformation {
    ServiceInterface SendQuoteRequest {
    ServiceInterface SendPO {
    ServiceInterface SendEstimate {
    ServiceInterface PaymentToSeller {
    ServiceInterface ReqeustShipping {
    ServiceInterface PaymentToShipper {
    Participant buyer {
    Participant seller {
    Participant shipper {
    Message RFI
    Message Catalog
    Message QuoteRequest
    Message Quote
    Message PO
    Message POConfirmation
    Message ShippingRequest
    Message ShippingCostEstimate
    Message ShipmentInformation
    Message Products
    Message SellerPayment
    Message ShipperPayment
    ServiceContract RequestForInformation "" among {
    ServiceContract SendQuoteRequest "" among {
    ServiceContract SendPO "" among {
    ServiceContract PaymentToSeller "" among {
    ServiceContract SendCatalog "" among {
    ServiceContract SendQuoteResponse "" among {
    ServiceContract AcceptPO "" among {
    ServiceContract ReqeustShipping "" among {
    ServiceContract ReportToBuyer "" among {
    ServiceContract PaymentToShipper "" among {
    ServiceContract SendEstimate "" among {
    ServiceContract ShipProducts "" among {
}
```

Figure 8: Sample Generated Service Architecture

10 SERVICE TO PROCESS TRANSFORMATION

In this section we will start with service model definition. The first step is to define the grammar for ServiceDSL that implements SoaML's core concepts, which are Service Architecture, Service Contract, Participant and Service Interface. Again, here is a portion of the grammar definition (more than ten elements are not shown in Fig. 9).

This DSL is simple enough to cover the structural aspects of SoaML model, and we even tried to include behavioral (sequence diagram) aspect in the grammar.

Next thing is to create a service model based on this DSL, which is shown in Fig. 10, which is done also by typing

The previous Fig. 2 showed a graphical representation of a sample Service Architecture at very high level. In the textual ServiceDSL model of Fig. 10, we included major elements under Service Architecture, because Service Architecture works as a root of the model in this language. Each usage of the typical language element is shown at least once, but not all the elements are shown by using the folding option to make the Figures smaller to fit in this paper. Also note that message sequencing is specified in Service Contract, e.g. with "buyer -> seller RFI optional," implementing message flows, or cross-lane object nodes, described in Fig. 1.

```
Model:
    (elements+=Type)*;
Type:
    DataType | SOACollaboration  ;
DataType:
    'type' name=ID;
FQN:
    ID ('.' ID)*;
SOACollaboration:
    'ServiceArchitecture' name=ID '{'
        (soaInterfaces+=SOAInterface)*
        (soaParticipants+=SOAParticipant)*
        (soaChannels+=SOAChannel)*
        (soaMessages+=SOAMessage)*
        (soaContracts+=ServiceContract)*
    '}' ;
SOAInterface:
    'ServiceInterface' name=ID '{'
        interfaces+=Interface*
    '}' ;
Interface: ProvidedInterface | RequiredInterface ;
ProvidedInterface:
    'ProvidedInterface' name=ID ('{'
        (params+=Param)*
    '}')? ;
RequiredInterface:[..]
Param:
    'Parameter' dir=Direction name=ID ':' type=[DataType] ;
SOAParticipant :
    'Participant' name=ID '{'
        (interactionPoints+=InteractionPoint)*
    '}'  ;
InteractionPoint:
    ServicePoint | RequestPoint ;
ServicePoint:
    'ServicePoint' name=ID ':' ref=[SOAInterface|ID] ;
RequestPoint:[..]
ServiceContract:
    'ServiceContract' name=ID description=STRING 'among' '{'
        roles+=Role*
        sequences+=Sequence*
    '}' ;
```

Figure 9: Partial ServiceDSL grammar

The major difference between this service model description and the previous process model description is in the style of control flow description, i.e. sequence versus activity, and the number of parties involved, i.e. only two parties in service model vs. possibly more than three parties in process model. In service model, service is the central concept and therefore major players in service model are consumer and provider. On the other hand, in process model, the focus is on control flow, object flow, conditional or parallel split and merge, covering all the players that could be more than three players. It is clear that service model is not able to express e.g. control flow within the same lane in process model, since they are not exposed as service interaction and of no interest in service model.

We can still apply model transformation to see what we can get even though the limitation is clear. Figure 11 shows a sample Xpand template to transform service model to process model. Figure 12 shows a transformed sample process based on the service model (Figure 10). It seems step portions of the process were successfully generated. But these are just concatenation of the sequences from Service Contract's sequence definitions.

If full control flow needs to be generated from the service description, process oriented description should be a part of the service model. In SoaML specification, these process aspects are treated as requirements specification to services, and therefore they are outside the scope of SoaML language itself (no stereotype or reference is defined as mandatory against Activity). The authors are planning to submit comments to OMG (or ISO if it is proposed) to clarify and enhance the standard or specification.

```
ServiceArchitecture BuySellShip {
    ServiceInterface BuyInterface {
        RequiredInterface RFI {
            Parameter in RequiredProducts : String
            Parameter out Information : Document
        }
        RequiredInterface PO {[..]
        RequiredInterface PaySeller {[..]
    }
    ServiceInterface ReceiveProduct {[..]
    ServiceInterface SellInterface {[..]
    ServiceInterface ShipperInterface {[..]
    Participant Buyer {
        RequestPoint RFI : BuyInterface
        RequestPoint PO : BuyInterface
        RequestPoint Payment : BuyInterface
        ServicePoint Products : ReceiveProduct
    }
    Participant Seller {[..]
    Participant Shipper {[..]
    ServiceChannel BuySell {[..]
    ServiceChannel SellShip {[..]
    ServiceChannel ShipBuy {[..]
    Message RFI
    Message Catalog
    Message PO
    Message POConfirmation
    Message RequestShipping
    Message EstimateForShipping
    Message ShippingStatus
    Message DeliverProduct
    Message SellerPayment
    Message ShipperPayment
    ServiceContract PlaceOrder "Ordering Products" among {
        consumer buyer performed by Buyer
        provider seller performed by Seller
        buyer->seller RFI optional
        buyer<-seller Catalog
        buyer->seller PO
        buyer<-seller POConfirmation
        buyer->seller SellerPayment
    }
    ServiceContract RequestShipping "Seller request for shipment" among {[..]
    ServiceContract DeliverProducts "Shipper delivers products" among {[..]
}
```

Figure 10: Sample Service Architecture

```
«IMPORT jp::ac::fun::xtext::soa::soaDsl»
«EXTENSION templates::Extensions»
«REM» Generating Process Steps «ENDREM»
«DEFINE main FOR SOACollaboration-»
«FILE name+".process"-»
process «this.name» {
«FOREACH this.soaParticipants AS participant-»
    role «participant.name» {
«FOREACH this.soaContracts AS contract-»
«FOREACH contract.sequences AS seq-»
«IF seq.consumer.performer==participant-»
    «IF seq.arrow=="->"-»
        step Request«seq.message.name»
    «ELSE-»
        step Receive«seq.message.name»
    «ENDIF-»
«ELSEIF seq.provider.performer==participant-»
    «IF seq.arrow=="->"-»
        step Receive«seq.message.name»
    «ELSE-»
        step Request«seq.message.name»
    «ENDIF-»
«ENDIF-»«ENDFOREACH-»«ENDFOREACH-»
    }
«ENDFOREACH-»
    role Artifacts {
«FOREACH this.soaMessages AS msg-»
        artifact «msg.name»
«ENDFOREACH-»
    }
}
«ENDFILE»«ENDDEFINE»
```

Figure 11: Sample model transformation definitions

```
process BuySellShip {
    role Buyer {
        step RequestRFI
        step ReceiveCatalog
        step RequestPO
        step ReceivePOConfirmation
        step RequestSellerPayment
        step RequestShippingStatus
        step ReceiveDeliverProduct
    }
    role Seller {
        step ReceiveRFI
        step RequestCatalog
        step ReceivePO
        step RequestPOConfirmation
        step ReceiveSellerPayment
        step RequestRequestShipping
        step ReceiveEstimateForShipping
        step RequestShipperPayment
    }
    role Shipper {
        step ReceiveRequestShipping
        step RequestEstimateForShipping
        step ReceiveShipperPayment
        step ReceiveShippingStatus
        step RequestDeliverProduct
    }
    role Artifacts {
        artifact RFI
        artifact Catalog
        artifact PO
        artifact POConfirmation
        artifact RequestShipping
        artifact EstimateForShipping
        artifact ShippingStatus
        artifact DeliverProduct
        artifact SellerPayment
        artifact ShipperPayment
    }
}
```

Figure 12: Sample Transformed Process

11 EVALUATION

For this specific buy-sell-ship example, we counted lines of text for each models. Manually written ProcessDSL-based model contained 227 lines, and ServiceDSL-based model transformed from it contained 88 lines. Manually written ServiceDSL-based model contained 125 lines, and ProcessDSL-based model transformed from it contained 40 lines. If both manually written models are semantically equivalent or very close, each transformed model should be reasonably compatible with the other manually written model. The line numbers comparison rates are 70.4% for Process to Service transformation, and 17.6% for Service to Process transformation. Although line by line comparison is better, this gives an implication that Process to Service transformation works much better than the other way. For instance, a portion of manually written ServiceDSL-based model below (Figure 13) is compared by a portion of generated model from ProcessDSL-based model (Figure 14).

```
ServiceContract PlaceOrder "Ordering Products" among {
    consumer buyer performed by Buyer
    provider seller performed by Seller
    buyer->seller RFI optional
    buyer<-seller Catalog
    buyer->seller PO
    buyer<-seller POConfirmation
    buyer->seller SellerPayment
}
```

Figure 13: Manually written Service Contract

```
ServiceContract RequestForInformation "" among {
    consumer buyerAsConsumer performed by buyer
    provider sellerAsProvider performed by seller
    buyerAsConsumer->sellerAsProvider RFI
}
ServiceContract SendQuoteRequest "" among {
    consumer buyerAsConsumer performed by buyer
    provider sellerAsProvider performed by seller
    buyerAsConsumer->sellerAsProvider QuoteRequest
}
ServiceContract SendPO "" among {
    consumer buyerAsConsumer performed by buyer
    provider sellerAsProvider performed by seller
    buyerAsConsumer->sellerAsProvider PO
}
ServiceContract PaymentToSeller "" among {
    consumer buyerAsConsumer performed by buyer
    provider sellerAsProvider performed by seller
    buyerAsConsumer->sellerAsProvider SellerPayment
}
ServiceContract SendCatalog "" among {
    consumer sellerAsConsumer performed by seller
    provider buyerAsProvider performed by buyer
    sellerAsConsumer->buyerAsProvider Catalog
}
ServiceContract SendQuoteResponse "" among {
    consumer sellerAsConsumer performed by seller
```

Figure 14: Transformed/Generated Service Contract

This shows that if there is good semantic relationship, we can get reasonable transformation result. The result will depend on how much of such good semantic relations with direction, e.g. Process to Service, exist between the two architecture styles.

12 RELATION WITH DISTRIBUTED COMPUTING

If we look at both process oriented and service oriented models in the context of distributed computing, they can be considered as candidate sources of execution on the platform, which may be internal enterprise systems or hybrid with cloud computing platform environment or full public cloud platform.

There is a category of technology called process engines that interpret and execute process definitions. Workflow engines are also considered as ones in this category. Their focus, however, is on controlling and monitoring the given process flow, and not on the execution of distributed services. Regarding SOA, Web Services is one of the typical implementation technologies, and they can be considered as base technology for distributed computing. With the use of SoaML, most of the necessary information to map down to SOA implementations is included in the model, and therefore it is not surprising to find products to do code generation based on SoaML model and actually run on the SOA runtime platform.

Our interest here is how close to implementation we can get based on process model via service model. We created a transformation template (not included in this paper) to generate skeleton interface codes of the service components. The result is shown in Fig. 15. As expected, there is not much detailed information included, since some of the control information is discarded when converting it to service model. To make this code work, it needs to be completed with more detailed information with implementation classes, WSDL files, and frameworks for SOA such as eclipse SOA Platform etc.

```
package jp.ac.fun.xtext.services;

import javax.jws.*;

public interface SendCatalogService {
    public void SendCatalog (
    );
}
public interface AcceptPOService {
    public void AcceptPO (
    );
}
public interface RequestForInformationService {
    public void RequestForInformation (
    );
}
public interface SendPOService {
    public void SendPO (
    );
}
public interface SendEstimateService {
    public void SendEstimate (
    );
}
public interface PaymentToSellerService {
    public void PaymentToSeller (
    );
}
public interface ReqeustShippingService {
    public void ReqeustShipping (
    );
}
public interface PaymentToShipperService {
    public void PaymentToShipper (
    );
}
```

Figure 15: Generated Skeleton Code

13 CONCLUSIONS

DSLs are usually used at area close to programming. We demonstrated that DSLs, which captures only core concepts of the target modeling language, can be applied to architecture descriptions that are Process-oriented and Service-oriented architectures and can be used to examine the difference between the two styles of modeling presented in this paper.

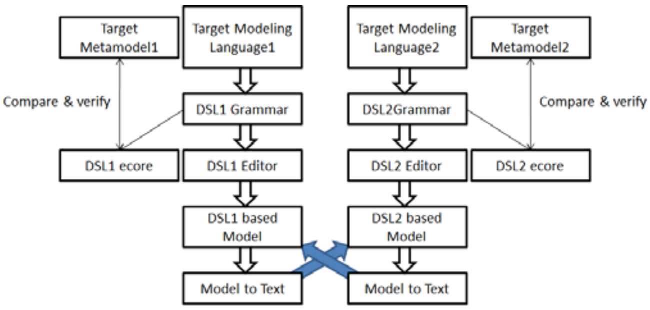

Figure 16: Model Transformation using M2T transformation

We created textual DSLs, including ProcessDSL and ServiceDSL, and showed sample transformations from process model to service model, and service model to process model. In doing so, we found a major difference between process modeling and service modeling. Something internal in process modeling will be lost when it is converted into service model, e.g. internal control flow. Orchestration of all the participants, which is the essential part of process model, is not possible in most cases when transforming service model to process model, since services are only meaningful to consumers and providers, i.e. between two parties, and normally orchestration aspects are left to higher level activities.

There is also a fundamental difference between the two, which is about level of abstraction. In process modeling, the level of abstraction is at end users or at business analysts level, but service modeling further includes interface specifications that are at architects' or developers' viewpoint. This caused transformation loss from service to process, and also was the reason of insufficient output from process model.

The possibility of service interface generation from process model was examined and only skeleton interface codes were generated because of the semantic gap between the two models with associated information loss in transformation. This does not, however, preclude the possibility of code generations from process model into process engines' environment and from service model into SOA environment.

Based on above, authors believe that they are showing the different aspects of the business system, and if mixed use is required, positioning process model as higher than service model will better work in enterprise architectures than positioning them in the opposite order or placing them at the same level.

Now, let us consider cost/performance of this project. The use case is an enterprise project to integrate its BPM based system with its SOA based system from multiple vendors or from multiple departments or by the result of M&A. If we apply the transformation template to this BPM based system, which is just automated transformation, we can get a list of candidate services required to implement the functionality of the BPM-based system. If this list is compared with existing SOA based system's services list, it is very likely to find similarities and missing pieces necessary for new integrated system. The merit would become clearer with the system size grows. Even though initial investment is required, this type of research will bring actual benefits in this kind of use cases.

Regarding the tooling, eclipse/Xtext provided necessary DSL development environment, and integrated model to text transformation facility Xpand/Xtend worked well to generate text from the DSL based models.

14 FUTURE WORKS

There are some areas where we need further works.

We will need to investigate a mechanism to verify the created DSLs, a mechanism to store trace/log information in transformation, more resources like practical examples around DSLs and model transformations, and more specialized tools to achieve specific activities.

We will also need to experiment on minimizing transformation loss such as internal steps in a process, possibly by introducing control flow manager in each lane so that each step could be transformed to internal service etc. or by introducing service orchestration function to the service model.

We are planning to use full RM-ODP model as source in the next experiment to see how component definitions in computational specification could contribute to process to service model transformations.

ACKNOWLEDGEMENTS

The authors thank contributors to eclipse modeling projects, especially to eclipse/Xtext open source project, and further thanks OMG and contributors of many modeling standards including UML, BPMN and SoaML. The authors also thank many people from various countries involved in ISO/IEC JTC1/SC21/WG7, SC7/WG19, and ITU-T SG19's continuing efforts to standardize and maintain RM-ODP family of standards.

REFERENCES

[1] J. A. Zachman, ``A Framework for Information Systems Architecture,'' IBM Systems Journal, Vol.26, No.3, p.276 (1987).

[2] FEA Consolidated Reference Model Document Version 2.3, http://www.whitehouse.gov/sites/default/files/omb/assets/fea_docs/FEA_CRM_v23_Final_Oct_2007_Revised.pdf (2007).

[3] OMG, ``Business Process Model and Notation (BPMN) Version 2.0,'' http://www.omg.org/spec/BPMN/2.0/PDF/ (2011).

[4] OMG, ``Semantics Of Business Vocabulary And Business Rules (SBVR), Version 1.0,'' http://www.omg.org/spec/SBVR/1.0/PDF/ (2008).

[5] D. Luckham, ``The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems,''Addison-Wesley (2002).

[6] OMG, ``Unified Modeling Language (UML),'' http://www.omg.org/spec/UML/ (2011).

[7] ISO/IEC 10746-2:2009, Information technology -- Open distributed processing -- Reference model: Foundations

ISO/IEC 10746-3:2009, Information technology -- Open distributed processing -- Reference model: Architecture

ISO/IEC 15414:2006, Information technology -- Open distributed processing -- Reference model -- Enterprise language

ISO/IEC 19793:2008, Information technology -- Open Distributed Processing -- Use of UML for ODP system specifications.

[8] M. Fowler, ``Domain-Specific Languages,'' Addison-Wesley (2011).

[9] A. Kleppe, ``Software Language Engineering, '' Addison- Wesley (2009).

[10] OASIS, ``Reference Model for Service Oriented Architecture 1.0,'' http://docs.oasis-open.org/soa- rm/v1.0/soa-rm.pdf (2006).

[11] OMG, ``Service oriented architecture Modeling Language (SoaML) Specification,'' http://www.omg.org/spec/SoaML/1.0/ (2012).

[12] OMG, ``Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification,'' http://www.omg.org/spec/QVT/1.1/PDF/ (2011).

[13] F. Jouault, F. Allilaire, J. Bezivin and I. Kurtev, ``ATL: A model transformation tool,'' Science of Computer Programming, Vol.72, No.1-2, Special Issue on Second issue of experimental software and toolkits (EST), pp.31-39 (2008).

[14] Eclipse, ``ATL Transformations,'' http://projects.eclipse.org/projects/modeling.mmt.atl.

[15] OMG, ``MOF 2 XMI Mapping Specification,'' http://www.omg.org/spec/XMI/ (2011).

[16] Eclipse Foundation, Xtext - Language Development Made Easy!, http://www.eclipse.org/Xtext/.

[17] The Spoofax Language Workbench, http://strategoxt.org/Spoofax.

[18] JetBrains : Meta Programming System - Language Oriented Programming environment and DSL creation tool, http://www.jetbrains.com/mps/.

[19] Xtext Users Japan, http://sites.google.com/site/xtextusersjapan/files.

(Received February 27, 2012)
(Revised February 28, 2013)

Akira Tanaka received M.E. from University of Tokyo, Japan in 1978. From 1978 to 2008, he was with Hitachi Ltd. His current research area includes model-based software development. He is a member of ACM, IEEE Computer Society and IPSJ.

Osamu Takahashi received master's degree from Hokkaido University in 1975. He is currently a professor at the Department of System Information Science at Future University Hakodate, Japan. His research interest includes ad-hoc net-work, network security, and mobile computing. He is a member of IEEE, IEICE, and IPSJ.

The Development of Database System for Route Recommendation based on Sensor Data

Yoshihiro Ito†, Yoshitaka Nakamura‡, Yoh Shiraishi‡, and Osamu Takahashi‡

†Graduate School of Systems Information Science, Future University Hakodate, Japan
‡School of Systems Information Science, Future University Hakodate, Japan
{g2111003, y-nakamr, siraisi, osamu}@fun.ac.jp

Abstract - In recent years, we have become able collect granular sensor data through many sensors attached to a sensor network in the real world. This technology has been used for urban sensing. It is expected to achieve new personal navigation on the basis of sensor data. However, to recommend routes on the basis of sensor data, we need to handle road data and sensor data, but it is difficult for only the application side to handle these data. Therefore, a database system has been developed that can map sensor data to the road data by using spatial interpolation and provide path planning results to a navigation application. In addition, the effectiveness of the proposed system is discussed.

Keywords: sensor network, urban sensing, database system, spatial interpolation, route recommendation

1 INTRODUCTION

In recent years, miniaturization and upgrading of the sensor devices have been miniaturized and upgraded further. Because of this, we can gather environmental information in the real world easily. These data are measured at specified time intervals and stored in a database for applications. For experiment, there have been attempts that gather environmental information in urban areas. This is called urban sensing[1].

Urban sensing is expected to realize new personal navigation based on sensor data. For example, by sensing local crowded and high temperature places preliminarily, it is possible to direct people to a convenient route that avoids such places[2].

To recommend route on the basis of sensor data, we need to map sensor data to road data. At this time, sensors need to be searched for in the sensors in the vicinity of the road. However, if no sensor is in the neighborhood, it is difficult to find the route based on sensor data. In addition, if sensor data and road data are managed in different data format, it is difficult for the application side to inquire and process data.

For these reasons, in this study, we developed a database system that can map sensor data to the road data by using spatial interpolation and provide results of route recommendation to a navigation application. In addition, we discuss the effectiveness of our proposed system.

2 RELATED WORK

This section discusses research related to our study. Specifically, we discuss the research of a sensor database system that provides interpolated sensor data for application in section 2.1. In addition, we discuss research on urban sensing in section 2.2 and research on navigation and path planning in section 2.3.

2.1 Sensor database system

To use sensor data collected by a sensor network, they must be processed. There is the technology that can do this processing in the field of database. This technology considers the sensor network as a virtual database to which it sends the query. There is some research into this technology[3]. This technology needs to understand the inquiries between the application and sensor network. Specifically, a function is required that accesses sensor network and feeds back the answer to an inquiry to the application user. This system is called the sensor database system[4].

There has been research into a sensor database system to feed back interpolated sensor data. Iwai et al.[5] and Ishii et al.[6] suggested a system that generates mesh structured sensor data by using spatial interpolation and estimate sensor data in each areas. However this system does not estimate sensor data on roads necessary for route recommendation.

In addition, Arai et al.[7] suggested the database system for time series interpolation query. When sensor data which application requires does not exist, this system can estimate sensor data at arbitrary time by using cubic spline interpolation. However this system does not estimate sensor data anywhere else.

2.2 Urban sensing

We provide examples of research into urban sensing. Takagi et al.[8] and Tobe and Kurata[9] gathered environmental information in urban areas by using a sensor network. In these studies, they used a fine-grained sensor network within a 600m radius in front of Tatebayashi City Station in Gunma, Japan. This sensor network consists of leaf node and sink node. The leaf node gathers sensor data and transmits them to the sink node. The sink node transmits sensor data to database.

In addition, CitySence[10] is a sensor network that monitors meteorological phenomenon and aerial pollution in urban area. This system expands a sensor node's battery life by connecting the sensor node to a street lamp. By this approach, City Sense is used for long-range research. For example, it can monitor environmental full-time "Public health" means how healthy people are.

2.3 Path planning

2.3.1. Basics of path planning

In this section, we discuss basic elements of path planning. Solving the optimum route problem requires a graph structure like that in Fig. 1.

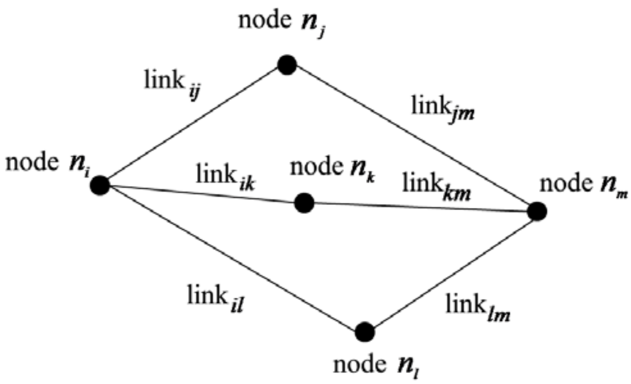

Figure 1: Graph structure

Each node is part of network that having information of node adjacent to them. In such structure, nodes are regarded as intersection, and the link as roads. This expresses the road network data. In addition, optimum routes can be searched for by giving the road distance to link as weight. Therefore, by changing one weight to another and paths can be planned on the basis of various parameters.

2.3.2. Related work on path planning

Architecture called pgRouting can search for optimal routes over the database[11]. This architecture can search for optimal routes by using a database table consisting of road data. This table needs a fixed-record format. It also needs "road id", "starting node", "end node", and "cost". However, pgRouting needs a table of road data. Therefore, when using sensor data as cost of the road data, a table must be created that records the road data for path planning, and updates it constantly during path planning. Applications have trouble executing these processes alone.

In addition, Endo and Tamura[12] suggested system using unusual WebAPI for navigation on the basis of sensor data. This system obtains sensor data by using this WebAPI, and calculates sensor data on roads, and searches route on the basis of these sensor data. This system is similar to our target system. The difference is that this system does not use special interpolation.

3 REQUIREMENTS OF PROPOSED SYSTEM

In this study, we aim to develop a system to recommend routes on the basis of sensor data. Therefore, we discuss its requirements below.

First, sensor data must be mapped to road data as cost. At this point, sensors need to be searched for in the neighborhood of the road. It makes no sense at all that there is sensor node on all roads. Therefore, an interpolate method is need to map sensor data to the road. In addition, our proposed system needs to calculate cost for route recommendation, to manage sensor data and road data at once and process inquiries from the application.

4 PROPOSED METHOD

4.1 Summary of proposed system

On the Basis of these considerations, we propose a database system to solve these problems. The database system needs the following functions.

A) Map sensor data to road data
B) Calculate sensor data-based cost of each road
C) Consolidate road data and sensor data
D) Process route recommendations
E) Process Inquiries

These functions are divided into two sections namely the Data Management Section which management various data and Database Management Section which recommends routes of the basis of sensor data (Figure 2).

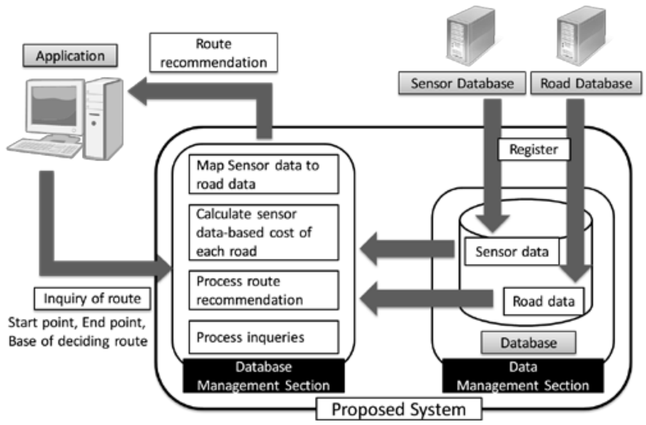

Figure 2: Summary of proposed system

The Data Management Section receives and manages sensor data and consolidates both road and sensor data. Road data are managed in road segment that separate each road from intersections. The Database Management Section receives sensor and road segment data and map sensor data to road segments. In addition, the Database Management Section handles and calculates the cost of road segment from mapped sensor data. Also, the Database Management Section processes inquiries and recommends route.

4.2 Management of road and sensor data

In this section, we discuss how to manage sensor and road data. Road data are managed as road segments that separate each road with intersections. Road segments are managed in the database as information of links described in section 2.3.1. Road segments are managed in the Road Segment Table which has the following data format (Table 1).

Table 1: Format of Road Segment Table

Attribute Name	Detail
Road ID	Identification numbers of road segment
Starting Node	Starting node numbers of road segment
End Node	End node numbers of road segment
Distance	Length of road segment (metric data)
Geometry	Location information of road segment (Line data)

The sensor data are managed in the Sensor Node Information Table and Sensor Data Table which has following data format (Table 2 and 3).

Table 2: Format of Sensor Node Information Table

Attribute Name	Detail
Sensor Node ID	Identification numbers of sensor node
Sensor Type	Types of sensor in sensor node
Geometry	Location information of sensor node (Point data)

Table 3: Format of Sensor Data Table

Attribute Name	Detail
Time	Identification numbers of sensor node
Data	The data which sensor node collected

Each of the Sensor Data Table named "Sensor Type_Sensor Node ID". In this way, we can access to Sensor Data Table by obtaining the Sensor Node ID and the Sensor Type from the Sensor Node Information Table.
The Data Management Section manages sensor data and road data as above, and the Data Management Section processes route recommendation by referring to all data in the Data Management Section.

4.3 Mapping sensor data to road data

Various factors that determine the environment of urban areas include buildings, parks and trees all of which affect the road environment. For example, when using temperature data, part of a road in the shadow of building is cooler than a part that is not. Therefore, to recommend the right route, sensor data need to be mapped on the road in a fine-grained manner. We explain how to do this.

First, we divide a road segment on the basis of its length. This is called a "divided road segment" (Figure 3). The number of divisions *x* is decided by the length of longest road segment in the road network (*max_length*) and defined width in meters is called "divide width" (*d*). Specifically, the number that is divided (*i*) is calculated as follows:

$$d \cdot (i-1) < x \leq d \cdot i \;\; (d \cdot i \leq \max\_length, i \geq 1) \qquad (1)$$

If the value of *i* is 1, the road segment stays undivided. In addition, the relationship between the road segment and divided road segment is as follows:

$$R_A = \{div_1, div_2, div_3, \ldots, div_i\} \qquad (2)$$

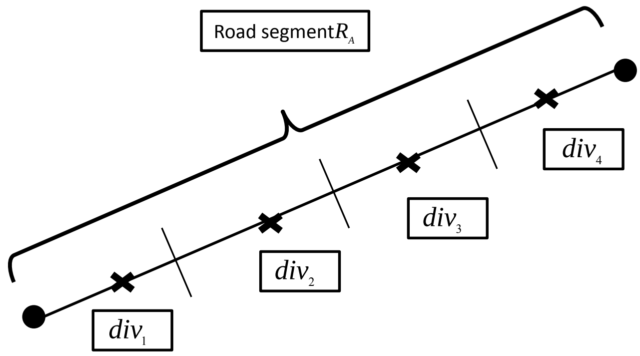

Figure 3: Divided road segment

By mapping sensor data to divided road segment generated by this method, we can know the environmental information on the road segment. However, depending on the situation of the sensor installation state, no sensor is considered to be near the road segment. In this case, possibly no sensor data are on the road segment, and this road segment cannot be used for route recommendation. In our proposed method, we solve this problem by using special interpolation. Specifically, we estimate sensor data on a road segment by using Inverse Distance Weighting(IDW)[13]. IDW is the method most often used by GIS analysts. It estimates unknown measurements as weighted average over the known measurements at nearby points, giving the greatest weight to the nearest points. By using IDW, we can know the environmental information on divided road segment from a sensor near them.

When using IDW, nearby points must be found. The proposed method solves this problem by apply the Variable Radius Method[14]. The Variable Radius Method is the method used to draw data points by describing a circle with a radius of decided distance around an unknown point and locate data points in this area. When the required number of data point does not exist in the area of the circle, the radius is expanded to search for more data points again (Figure 4).

With the above method, we explain the method for sensor data estimation on a divided road segment (div_1, div_2, …div_i). First, calculate the midpoint of divided road segments and set these midpoints as interpolate points. Next, define the search range in meters, and search for sensor node by using the Various Radius Method. Specifically, this method draws sensor node point by describing the circle with a radius of the search range introduced before around midpoint of a divided road segment. When no sensor node is in the area of circle, double the search range and search for sensor nodes again. It becomes possible to map sensor data on all divided road segments by processing above method to the midpoint of divided a road segment.

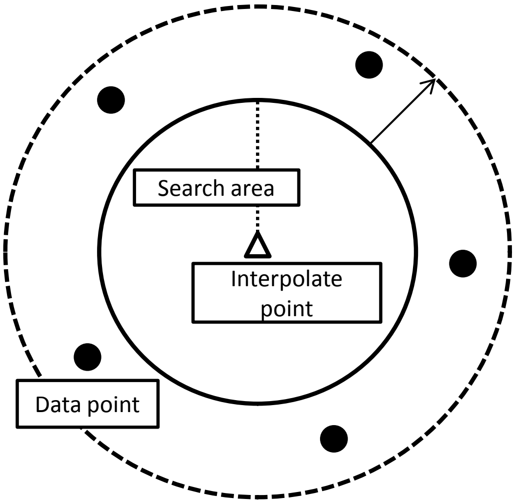

Figure 4: Variable Radius Method

4.4 Calculation of cost from sensor data

The method proposed in section 4.3, map sensor data on divided road segments. However, the cost needs to be calculated on each road segment for route recommendation. Therefore, we adapt the Dijkstra method which is the path planning algorithm because this algorithm excels in efficient and application, and define the cost-calculating formula for sensor data-based route recommendation. This formula calculates the cost of each road segment by using sensor value and the value which is decided on the basis of users feel comfortable (For example, using temperature, its value defined as 25 degrees Celsius). We explain this calculation formula below.

First, road segment R_A maps sensor data as shown in Fig. 5.

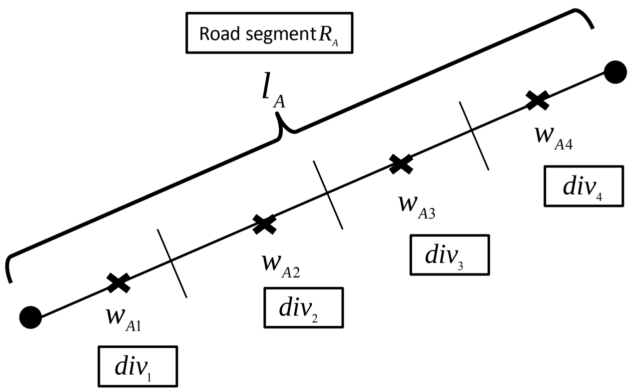

Figure 5: Parameter of road segment

w_{ik} is the estimated sensor data on divided road segment, and l_A is the length of the road segment R_A. In addition the division number of road segment is n, and the value which is decided on the basis of users feels comfortable as t_d. We find the sensor data based cost of each road segment by using these parameters. When sensor data and route length are considered, cost C_A is calculated by formula (3), and when only sensor data are considered, calculate cost C_A by formula (4).

$$C_A = \left\{ \sum_{k=1}^{n} \left| w_{Ak} - t_d \right| \right\} \cdot l_i \qquad (3)$$

$$C_A = \{ \max(w_{Ak}) - t_d \}^2 \qquad (1 \le k \le n) \qquad (4)$$

Formula (3)'s cost of road segment increases as the difference between t_d and w_{ik} increases. Therefore, by using this formula, we can select a convenient route for the user. In addition, when the length of a road segment is multiplied by the difference between w_{ik} and t_d, cost C_A can consider the route length. On the other hand, Formula (4) attaches importance to sensor data. It does not consider the route length but calculates cost of road segment from sensor data only. Next, we describe formula (3) the "sensor and distance base method" and formula (4) the "sensor base method".

4.5 Flow of processing

Figure 6 shows the processing flow with the above method.

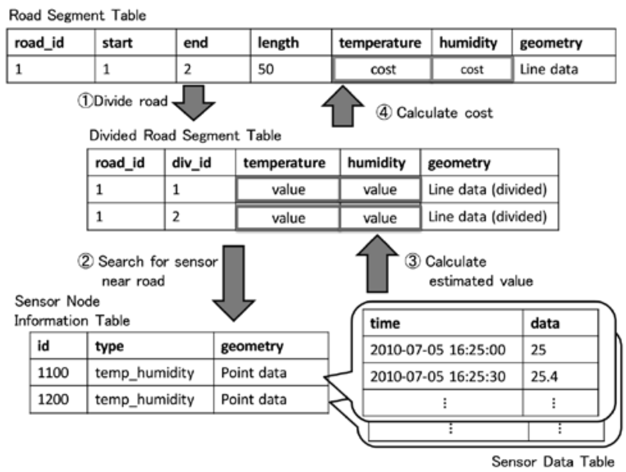

Figure 6: Flow of processing

First, the proposed method refers to the length of a road segment from the Road Segment Table and divides the road segment on the basis of the length. These results are stored in the Divided Road Segment Table (1). Next, to map interpolated sensor data onto a divided road segment, the method compares location information of the divided road segment and location information of sensor node, and pick up Sensor Node ID from Sensor Node Information Table by using the Various Radius Method (2). Therefore, by using Sensor node ID, the method searches for and obtains sensor data from Sensor Data Table, calculates interpolated sensor data on a divided road segment by using IDW and inserts these data into the Divided Road Segment Table (3). Finally, it calculates the cost for route recommendation by using the sensor and distance base method or sensor base method (4). After processing these methods, the proposed method recommends a route by referring to the cost in the Road Segment Table and applying Dijkstra method. This data structure can control different types of sensor.

5 EXPERIMENT AND DISCUSS

5.1 Experiment environment

To verify the effectiveness of the proposed system, we conducted an experiment by using one computer (CPU: 3.2GHz, Memory: 4GB, HDD: 500GB OS: Windows 7 Professional). We developed a database system that had the function we proposed on this computer. Also, this database system is implemented by Java Servlet and uses a database-management system of PostgreSQL8.4, and it uses PostGIS1.5 an advanced function of PostgreSQL to manage geometry data.

5.2 Experiment for evaluation

In our experiment, we used sensor data gathered by the sensor network in Tatebayashi City, Japan. This sensor network gathers temperature and humidity data and a 600m radius in front of the station, as shown in Fig. 7.

Figure 7: Distribution of sensor nodes

In our experiment, we used data gathered in summer, 2010. From these, we choose sensor data on clear days: August 23, 24 and 28, 2010. In addition, we used sensor data in August 21, 2010 because its temperature data on each point were high and low.

In our experiment, attempted to map and estimate sensor data. Specifically, we measured processing times elapsing from the beginning of the calculation to estimate the first to the last sensor data. In addition, we measured estimation accuracy in accordance with different times of day and evaluated the recommended routes. We discuss the experiment results below.

5.2.1. Route recommendation

We describe the result of route recommendation. In this experiment, we generated routes by using sensor and distance base method and sensor base method, and compared the generated and shortest routes. We used temperature data from 16:00 August 21, 2010. Figure 8 shows the results. Also, the value of temperature at which users feel comfortable (t_d) is defined as 25 degrees Celsius.

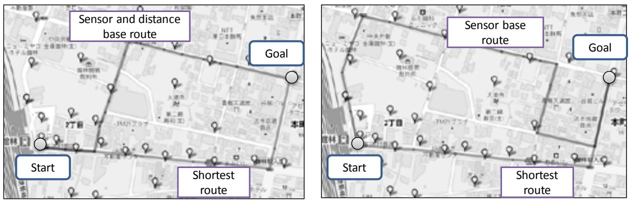

Figure 8: Generated route
(Left: Sensor and Distance Base Method
Right: Sensor Base Method)

To analyze these generated routes, we show a temperature gradient of road segment estimated by the proposed system in Fig. 9.

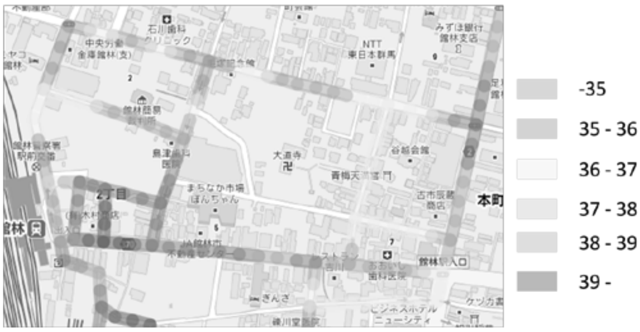

Figure 9: Temperature gradients of road segments

From this temperature gradient of a road segment, we define t_d as 25 degrees Celsius, and the proposed system was expected to recommend the coolest route. Figures 10, 11 and 12 show estimated temperature gradients of each route.

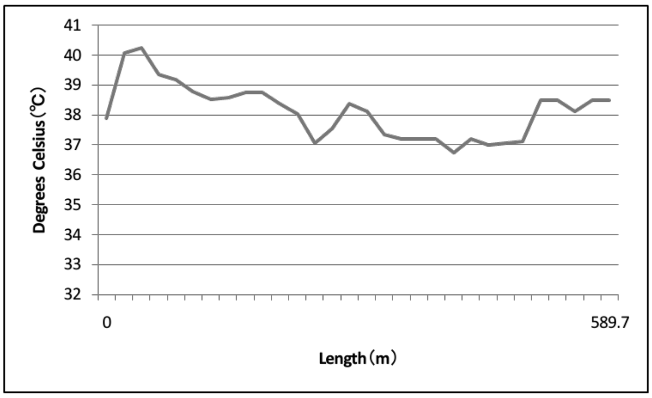

Figure 10: Estimated temperature gradient of shortest route

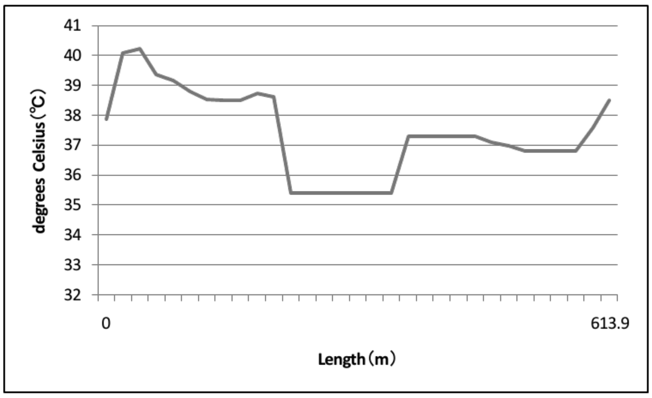

Figure 11: Estimated temperature gradient of sensor and distance base method route

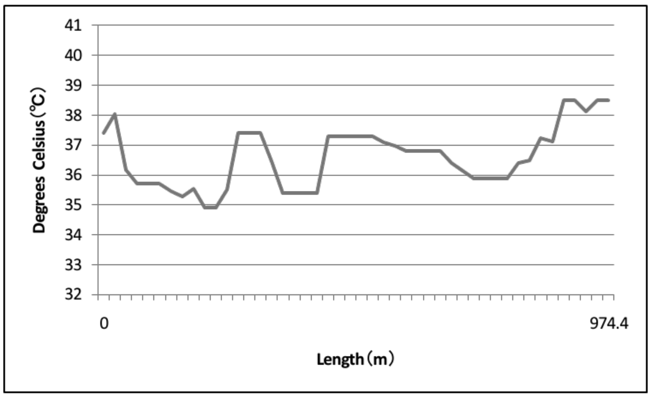

Figure 12: Estimated temperature gradient of sensor base method route

5.2.2. Processing time for estimating sensor data

We describe the processing time to calculate estimated value by using IDW. To use IDW, we define the value of the search range for the Variable Radius Method. We measure the processing time of value estimation by changing the search range. Specifically, we define the search ranges as 50, 100, 150, 200, 250, 300, 400, 450, 500, 1000 and 2000 meters, and measure processing time elapsing from the beginning of the calculation to estimate the first to the last sensor data of each range. These results are shown in Fig. 13. Also, divide width of road segment (d) is defined as 25 meters.

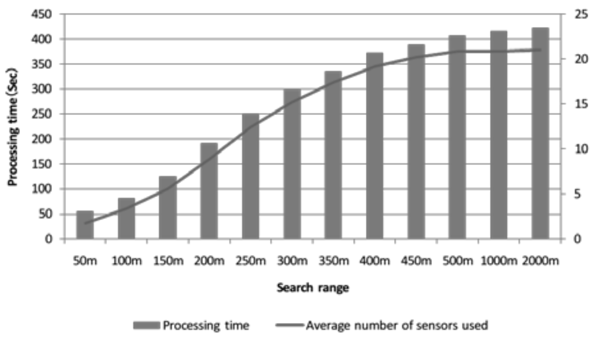

Figure 13: Processing time

According to these results, processing time increases as the search range expands, and the average number of sensor nodes used to calculate the estimated value at an interpolate point also increases. Processing time increases directly with the average number of sensor nodes used to calculate the estimated value, but when the search range is 500 meters, all sensor nodes are used to calculate all points, Neither processing time or the number sensor nodes significantly change after this.

5.2.3. Estimation accuracy

We describe results of an estimation accuracy experiment for the proposed system. This system has difficulty to analyzing estimation accuracy because it estimates sensor data on unknown points. For this reason, we set the place of a sensor node that has actual measured value in Tatebayashi City's sensor network as the interpolate point and estimated sensor value on that point by using proposed method. We tested this method on all points at which a sensor node existed, and calculated the difference between estimated and actual values. This difference was calculated by relative error and derived from the following calculation in percentage terms (E: estimated value, A: actual value):

$$\frac{|E-A|}{A}\cdot 100 \quad (5)$$

First, we measured estimation accuracy by changing the s earch range. Search ranges were the same as those given in 5.2.2', and we calculated relative error for all sensor node po ints by using formula (5) and calculated their mean. In this e xperiment, we used temperature data from 14:00 August 23, 24 and 28, 2010. Figure 14 shows the results.

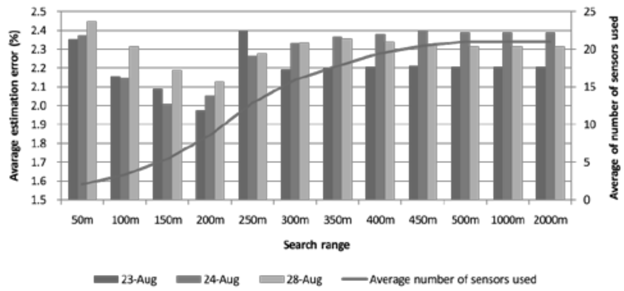

Figure 14: Change in estimated error by changing search range

As shown in Fig. 14, as the search range expanded from 50 to 200 meters, estimation error decreased gradually. However, estimation error is increased from 250 to 500 meter. That difference may have been caused by using sensor data far from the interpolating points for estimation calculation. Next, we measured the estimation accuracy for temperature in accordance with different times of day. The times of day were 8:00, 14:00 and 20:00, and we calculated the average estimation error for each. Figure 15 shows the results.

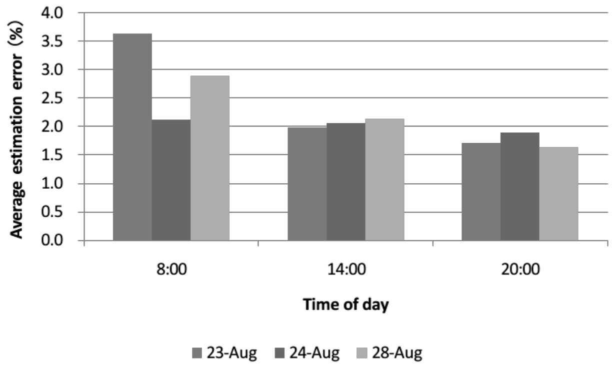

Figure 15: Estimation error by time of day

A 3% estimation error represented a degrees Celsius error. From this reason, Figure 15 shows the estimation error is small overall. However, at 8:00, estimation error was considerably larger on 14:00 and 20:00. To analyze this, we checked the estimated error of each estimated point at 8:00. Fig. 16, 17 and 18 show the results.

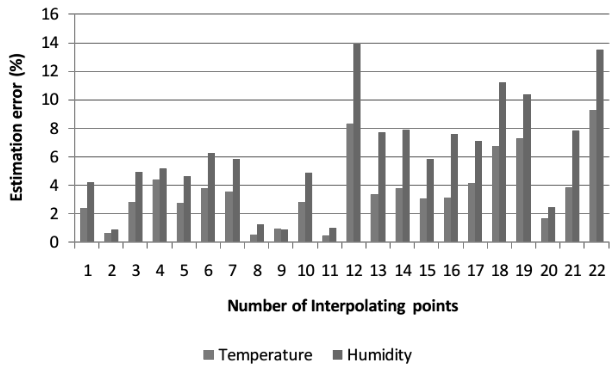

Figure 16: Estimation error at 8:00 Aug-23

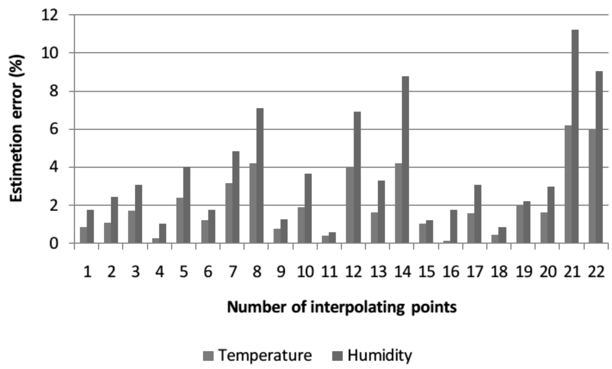

Figure 17: Estimation error at 8:00 Aug-24

In this experiment, we calculated estimation errors for temperature and humidity. These results show the estimation errors for some interpolating points are large. In addition, this figure shows the degree of estimation error for temperature tends to be the same as the degree of estimation error for humidity. For example, we found that at places at which large estimation error was measured for temperature, a large estimation error was similarly measured for humidity.

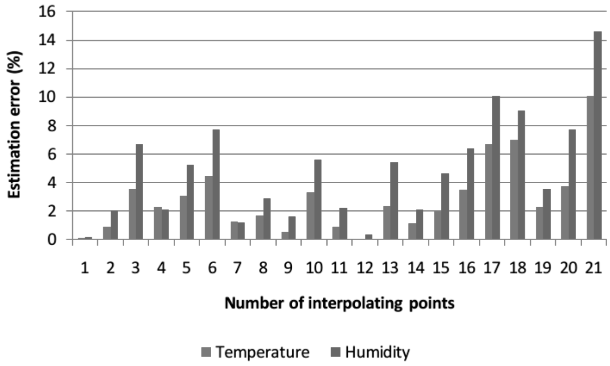

Figure 18: Estimation error at 8:00 Aug-28

From these results, the estimation accuracy apparently depends on a sensor node's installation location and situation. Therefore, we analyzed the interpolating points at which estimation error for temperature was over 5% at 8:00 August 23. Figure 19 shows the results. Sensor nodes with a cross have no data or error data, so they were not used to calculate the estimation.

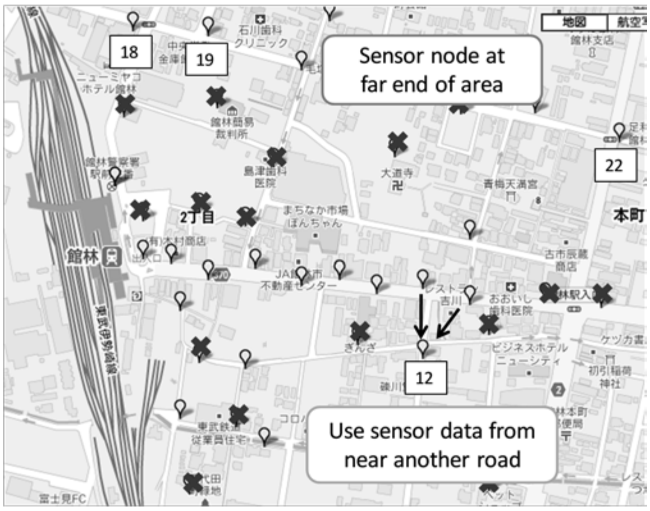

Figure 19: Interpolated point that measured large estimation error at 8:00 Aug-23

At 8:00 August 23, the installation location of sensor nodes the IDs of which ID is 18, 19, 22 and 12 had an estimation error over 5%. First, the standout feature of the interpolating point that had large errors was that they existed in the far end of the sensor network area. This applies to location of sensor node 18, 19 and 22. This reason is considered to be the lack of sensor nodes in far end of the area near these three. On the other hand, sensor node 12 exists near the center of the sensor network, but its estimation error was large. The reason is considered to be that no sensor node for which data are available is near sensor node 12, so it uses sensor node near another road to calculate estimations.

In another day's results, large error classifiable into these two patterns was measured at interpolating points. Therefore, the solution is needed to these problems.

5.2.4. Discussion of experiment results

In this section, we discuss results of the above experiment. In the evaluation of route recommendation, we confirmed that the proposed system recommend routes the way we assumed by using our proposed method and estimated sensor data.

However, from the evaluation of estimation accuracy, we found that some interpolating points have large estimation error. This error is possibly due to location and situation of interpolating points. This error affects the accuracy of recommended routes. To solve this problem, errors must be reduced at such points. One possible solution to this problem is participatory sensing[15]. By using participatory sensing, we can gather sensor data not only from a fixed sensor network but also from mobile devises. Therefore, estimation accuracy may be improved by gathering sensor data near point that have large estimation error by using participatory sensing.

In addition, from the evaluation of processing time, we determined that it takes time to estimate sensor data correctly. To solve this problem, processing must be speed up by improving database structure. Furthermore, the proposed system processes estimations and records this processing at constant intervals, and if an inquiry comes from the application, proposed system executes the route recommendation process only.

6 CONCLUSION

In this study, we developed a database system that can estimate sensor data on roads by using spatial interpolation and recommend routes on the basis of this sensor data. In addition, we evaluated processing time of the proposed system, estimation accuracy of sensor data and recommended routes generated by the proposed system.

In the future, to improve estimation accuracy of sensor data, we will consider a database system dealing with participatory sensing data. To achieve this system, we will consider a method to deal with mobile sensing data. Moreover, we will consider the way to use another sensor data.

REFERENCES

[1] T. Ono, H. Ishizuka, K. Ito, Y. Ishida, S. Miyazaki, O. Mihirogi and Y. Tobe, ``UScan : Towards Fine-Grained Urban Sensing,'' International Workshop on Real Field Identification (RFId 2007), pp.18-27 (2007).

[2] K. Kanai，T. Ono, I. Hiroki, K. Ito, Y. Sawa, O. Mihirogi and Y. Tobe, ``Proposal of an Urban Information Sensing System,'' IPSJ SIG Technical Report UBI, Vol.2008, No.18, pp.141-148 (2008). *(in Japanese)*

[3] S. Ando, Y. Tamura, Y. Tobe and M. Minami, ``Sensor Network Technology,'' Tokyo Denki University Press (2005). *(in Japanese)*

[4] Y. Shiraishi, ``Database Technologies for Sensor Networks,'' IPSJ Journal, Vol.47, No.4, pp.387-393 (2006). *(in Japanese)*

[5] M. Iwai, N. Thepvilojanapong, H. Ishizuka, Y. Nakamura, K. Kanai, Y. Shiraishi and Y. Tobe, ``TomuDB: A Database Management system for Distributed Urban Sensors,'' IEICE Technical Report USN, Vol.108, No.138, pp.13-18 (2008). *(in Japanese)*

[6] N. Ishii, Y. Shiraishi, H. Ishizuka and Y. Tobe, ``An Integration Method of Heterogeneous and Multi-resolutional Sensor Data Utilizing Spacial Interpolation,'' IPSJ Journal, Vol.49, No.SIG7 （TOD37）, pp.44-58 (2008). *(in Japanese)*

[7] K. Arai, Y. Shiraishi and O. Takahashi, ``Design and development of a sensor database system for time series interpolation query,'' The 72nd National Convention of IPSJ (3), pp.293-294 (2010). *(in Japanese)*

[8] A. Takagi, K. Sugo, Y. Ishida, T. Morita, T. Iwamoto, H. Kurata and Y. Tobe, ``A Practical Implementation of Micro-Climate Networked Sensing : Case at Tatebayashi City, Gumma,'' IEICE Technical Report USN, Vol.109, No.248, pp.13-18 (2009). *(in Japanese)*

[9] Y. Tobe and H. Kurata, ``A Practical Implementation of Micro-Climate Networked Sensing - Case at Tatebayashi City,'' IPSJ Journal, vol.51, No.6, pp692-699 (2010). *(in Japanese)*

[10] R. Murty, G. Mainland, I. Rose, A. R. Chowdhury, A. Gosain, J. Bers and M. Welsh, ``CitySense: A Vision for an Urban-Scale Wireless Networking Testbed,'' Proceeding of the 2008 IEEE International Conference on Technologies for Homeland Security (2008).

[11] pgRouting Project, ``pgRouting", http://www.pgrouting.postlbs.org.

[12] T. Endo and Y. Tamura, ``Research of Sensor Networks by Using Web API- : Development of Navigation System by Using Sensor Data,'' The 70th National Convention of IPSJ (3), pp23-24 (2008). *(in Japanese)*

[13] P. A. Longley, M. F. Goodchild, D. J. Maguire and D. W. Rhind, ``Geographic Information Science,'' John Wiley & Sons, Ltd (2001).

[14] C. Zhang, ``Analysis of Spatial Data by Using Geographic Information System", KOKON (2001). *(in Japanese)*

[15] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy and M. B. Srivastava. ``Participatory sensing,'' World Sensor Web Workshop, ACM Sensys (2006).

(Received February 27, 2012)
(Revised November 23, 2012)

Yoshihiro Ito received B.E degrees from Future University Hakodate in 2011. His research interests include database and GIS. He currently works at his master's degree in Graduate school of system information science, Future University Hakodate.

Yoshitaka Nakamura received B.E., M.S., and Ph.D. degrees from Osaka University in 2002, 2004 and 2007, respectively. He is currently a research associate at the School of Systems Information Science, Future University Hakodate. He is a member of IEEE and IPSJ.

Yoh Shiraishi received doctor's degree from Keio University in 2004. He is currently an associate professor at the Department of Media Architecture, School of Systems Information Science, Future University Hakodate Japan. His research interests include database, mobile sensing and ubiquitous computing. He is a member of IPSJ, IEICE, GISA and ACM.

Osamu Takahashi received master's degree from Hokkaido University in 1975. He is currently a professor at the Department of System Information Science at Future University Hakodate, Japan. His research interest includes ad-hoc net-work, network security, and mobile computing. He is a member of IEEE, IEICE, IPSJ.

Submission Guidance

About IJIS

International Journal of Informatics Society (ISSN 1883-4566) is published in one volume of three issues a year. One should be a member of Informatics Society for the submission of the article at least. A submission article is reviewed at least two reviewer. The online version of the journal is available at the following site: http://www.infsoc.org.

Aims and Scope of Informatics Society

The evolution of informatics heralds a new information society. It provides more convenience to our life. Informatics and technologies have been integrated by various fields. For example, mathematics, linguistics, logics, engineering, and new fields will join it. Especially, we are continuing to maintain an awareness of informatics and communication convergence. Informatics Society is the organization that tries to develop informatics and technologies with this convergence. International Journal of Informatics Society (IJIS) is the journal of Informatics Society.

Areas of interest include, but are not limited to:

- Computer supported cooperative work and groupware
- Intelligent transport system
- Distributed Computing
- Multi-media communication
- Information systems
- Mobile computing
- Ubiquitous computing

Instruction to Authors

For detailed instructions please refer to the Authors Corner on our Web site, http://www.infsoc.org/.

Submission of manuscripts: There is no limitation of page count as full papers, each of which will be subject to a full review process. An electronic, PDF-based submission of papers is mandatory. Download and use the LaTeX2e or Microsoft Word sample IJIS formats.

http://www.infsoc.org/IJIS-Format.pdf

LaTeX2e

LaTeX2e files (ZIP) http://www.infsoc.org/template_IJIS.zip

Microsoft Word™

Sample document http://www.infsoc.org/sample_IJIS.doc

Please send the PDF file of your paper to secretariat@infsoc.org with the following information:

Title, Author: Name (Affiliation), Name (Affiliation), Corresponding Author. Address, Tel, Fax, E-mail:

Copyright

Publisher

Address: Informatics Laboratory, 3-41 Tsujimachi, Kitaku, Nagoya 462-0032, Japan

E-mail: secretariat@infsoc.org

CONTENTS