A Frame Representation of User Requirements for Automated Data Visualization

Mitsunori Matsushita, Hayato Yonezawa, Tsuneaki Kato [1]

Abstract. We propose a new visualization system that automatically draws a graph from a large amount of numerical data corresponding to a user's requirement. The system has the following functions: to understand the user's requirement, to select the necessary data from a datatable, to convert and aggregate the selected data to a suitable granularity, and to visualize the data with an appropriate graphical style. In this paper, we introduce a semantic frame in order to formally deal with the user's requirement. The semantic frame consists of two subframes: one is mainly for selecting and restructuring data, and the other is mainly for choosing an appropriate graph-type and emphasizing focal characteristics. We also explain an aggregation algorithm for making the necessary datatable. With this system, a user only inputs his/her requirement described in a natural language, and then he/she is able to get an appropriate graph from an enormous amount of data even if the data granularity is not well suited to the requirement.

1 INTRODUCTION

Recently, the storage and processing speed of computers have been improved dramatically with the rapid progress in technology. These improvements have made it possible to transact operations with a wide variety and an enormous amount of data. Moreover, with the enrichment of network infrastructures, users can now share data with each other and use them according to their needs. However, it is still difficult for users to collect beneficial data from a large amount of stored data because: (1) each user's load increases according to the amount of data, (2) the appropriate graph depends not only on the characteristics of the data but also on the interests of each user, and (3) stored data is not always described in a granularity best suited to each user's requirement.

In order to solve this problem, a system that helps a user handle a large amount of data easily according to the interest of the user is needed. It is also important that the system have an intuitive and easy interface. Accordingly, we propose a new visualization system that has a natural language interface. The functions of the system are to understand the user's requirement, to select the necessary data from stored data, to convert and aggregate the selected data to a suitable granularity, and to visualize the data in an appropriate graphical style.

2 VISUALIZATION PROCESSES OF OUR SYSTEM

Our visualization system draws a graph suited to a user's requirement by the following three processes.

(A) Understanding the user's requirement

The purpose of this process is to extract information from the user's requirement that is necessary for data aggregation. Our system can recognize two types of user requirements as follows[2].

(ex1) *In 1995, the precipitation in the Kinki district was significant in Kyoto Prefecture.*

(ex2) *What was the population of each gender by prefecture in the Shikoku district in 1999?*

The user requirement (ex1) is an example of a declaration-type requirement, and (ex2) is a question-type requirement. A declaration-type requirement is used when the user wants to get a graph that is expected to support his/her thoughts, and a question-type requirement is used when the user wants to know the tendency of data.

From the viewpoint of drawing a graph, the difference between these requirements is that the former includes two sorts of clue information for visualization, such as information on "what to draw" and information on "focal point." The information on "what to draw" relates to decisions on drawing objects, whereas the information on "focal point" relates to decisions on the drawing method such as a graph-type or emphasis method. For instance, in the case of the user requirement (ex1), "In 1995," "the precipitation," and "the Kinki district" are information on "what to draw," and "significant in Kyoto Prefecture" is information on "focal point." Conversely, the latter type of requirement only includes information on "what to draw."

In this process, these two types of information are extracted from the user's requirement in the form of a semantic frame.

(B) Organizing the necessary data

The purpose of this process is to select the necessary tuples from a datatable, to convert them into a suitable granularity, and to aggregate them in compliance with the user's requirement. These operations are executed based on the semantic frame generated in process (A).

(C) Visualizing the data

In this process, the system makes a graph suitable for the user's requirement based on the organized datatable constructed in the previous process.

Our system determines candidates of appropriate graph-types based on the characteristics of the organized data and the semantic frames. Several graph candidates are chosen from 13 graph-type candidates in the order of preference.

Then, the system determines the necessary parameters for drawing a graph by considering how to emphasize focal characteristics of the graph (e.g., the scale of the Y-axis or the color scheme).

Finally, the system makes a graph by using these parameters.

Since our main goal is to clarify how to draw a graph that reflects

[1] NTT Communication Science Labs., Hikaridai 2-4, Seika-cho, Soraku-gun, Kyoto 619-0237 Japan, email: {mat, hayatoyo, kato}@cslab.kecl.ntt.co.jp

[2] Our current system can handle Japanese sentences only.

ECAI 2000. *14th European Conference on Artificial Intelligence*
Edited by Werner Horn
Published in 2000 by IOS Press

a user's requirement described in a natural language, we mainly describe the first two processes in this paper.

3 DEFINITION OF TARGET DATATABLE

In our system, all original data is stored in the following datatable.

Let A_i be an attribute of data and a_{ij} be an instance of A_i. The relationship between A_i and a_{ij} is described as follows.

$$dom(A_i) = \{a_{i1}, \ldots, a_{im}\}, \quad (1)$$

where the function dom is a mapping function between A_i and a_{ij}.

The datatable consists of a finite subset of Cartesian products of $dom(A_i)$. In other words, a tuple $\langle d_1, \ldots, d_n \rangle (d_i \in dom(A_i), 1 \leq i \leq n)$ means one datum in the datatable.

To simplify the problem, we assume that the datatable satisfies the second nominal form where $\{A_1, \ldots, A_{n-1}\}$ are candidate key attributes and only A_n is a non-key attribute. That is, all candidate key attributes are independent of each other (i.e., $A_i \cap A_j = \phi$ for $i \neq j$), and attributes whose instances belong to the same domains are not permitted even if they are described in a different granularity. For instance, A_i described in the granularity of "city" and A_j described in the granularity of "prefecture" are not permitted because they belong to the same domain and an implication relation exists between them. As a first step of our approach, we establish the restriction that the measure of A_n be numerical.

Table 1 shows an example of a datatable. In this datatable, "place" and "measuring date" are candidate key attributes and "precipitation" is a non-key attribute. In this paper, we assume that the user requirement (ex1) is a request to this datatable.

Table 1. Example of a datatable

Place	Measuring date	Precipitation
Hokkaido	1995/01/01 01:00	2
Hokkaido	1995/01/01 02:00	1
Hokkaido	1995/01/01 03:00	3
...	...	...
Okinawa	2000/01/27 22:00	2
Okinawa	2000/01/27 23:00	1

4 UNDERSTANDING A USER REQUIREMENT

In order to make a computer understand a user's requirement described in a natural language, it is necessary to convert the requirement into a formal representation that the computer can handle. Therefore, we propose a semantic frame as a formal representation of a user's requirement. The frame consists of two sub-frames: a "what to draw" frame and a "focal point" frame. Note that the "focal point" frame might not be used according to the type of user requirement. For instance, user requirement (ex1) is described by using both frames, whereas user requirement (ex2) is described by using the "what to draw" frame only.

4.1 "What to draw" frame

Drawing objects mentioned in a user's requirement are described in a "What to draw" frame. This frame is mainly used to aggregate a datatable in a later process.

This frame consists of five slots: VAR, ATTR, MEAS, GRANU, and REST. In this paper, *var*, ATTR(*var*), MEAS(*var*), GRANU(*var*), and REST(*var*) mean the value of each slot respectively. A "what to draw" frame has several lines depending on the user's requirement. Information about one drawing object is represented in one line. One drawing object relates to one attribute in a datatable. Therefore, the number of lines must not exceed the number of attributes in the datatable.

VAR	ATTR	MEAS	GRANU	REST
X	measuring date	date	year	x_1 = 1998
Y	place	nominal	prefecture	$y_i \in$ Kinki district
Z	precipitation	numerical	mm	

Figure 1. "What to draw" frame of user requirement (ex1)

VAR	ATTR	MEAS	GRANU	REST
W	measuring date	date	year	w_1 = 1997
X	place	nominal	prefecture	$x_i \in$ Shikoku district
Y	gender	nominal	gender	y_1 = man, y_2 = woman
Z	population	numerical	person	

Figure 2. "What to draw" frame of user requirement (ex2)

User requirements (ex1) and (ex2) are translated into the frames shown in Figure 1 and Figure 2, respectively. Our system extracts information that comes under each slot by matching each user requirement and generalized patterns of requirements stored in the system's rulebase. The patterns are mainly connection rules of auxiliary verbs and postpositional particles in Japanese.

The details of each slot are as follows.

- **VAR**
  A *var*, a slot value of VAR, is a temporary unique identifier.
- **ATTR**
  The ATTR slot takes the name of a drawing object as a slot value. The name of each drawing object corresponds to an attribute in a datatable. All drawing objects are determined by the noun phrases in the user's requirement.
  For instance, the noun phrases such as "precipitation," "1995," "the Kinki district," and "Kyoto Prefecture" appear in user requirement (ex1). These noun phrases correspond to attributes in the datatable (e.g., Table 1) such as "precipitation," "measuring date," and "place." Note that both "the Kinki district" and "Kyoto Prefecture" correspond to the same attribute "place." These attributes are accepted as ATTR(*var*).
- **MEAS**
  MEAS indicates the measure of the drawing object. It is used to determine an operator set that can be applied to the instance set of attributes corresponding to ATTR(*var*). The operator set depends on MEAS(*var*) and is used in process (B). In our system, "nominal," "date," "numerical," or "order" can be taken as MEAS(*var*). For instance, when MEAS(*var*) is "nominal," the instances of the corresponding attribute may have implication relations by lower/upper concepts. Therefore, the instances can be applied with coarser/finer granulation operations. However, they cannot be applied with a mathematical operation such as a plus or minus operation. In contrast, when MEAS(*var*) is "numerical," the instances of the corresponding attribute can be applied with a mathematical operator and can also be applied with a translation operator that translates the instances into ratio or cumulative values.

- **GRANU**
  GRANU is the granularity of the drawing object. The user may want to visualize the data in a different granularity from the stored data in the datatable. In order to absorb the granularity difference, our system aggregates the tuples to a suitable granularity based on GRANU(*var*).
  For instance, "place" of user requirement (ex1) is described in the granularity of prefecture. If the instances of attribute "place" in the datatable are described in the granularity of city, our system aggregates the data into a granularity of prefecture.

- **REST**
  REST indicates the restrictions of the drawing object mentioned in a user's requirement and is used to choose the necessary tuples from the datatable.
  For instance, in user requirement (ex1), as the user's interest is limited to the Kinki district as the "place," tuples concerning other districts such as the Kanto district are not necessary. In addition, the user's interest is also limited to 1995 for the "measuring date." In our system, REST(*var*) can be described in several ways such as instance designation (e.g., y_1 = Kyoto Prefecture), upper concept designation (e.g., $y_1 \in$ Kinki district), or region designation (e.g., $0 \leq y_i \leq 100$).

Our current system visualizes data on 2-dimensional statistical charts and the charts can be drawn if a $m \times n$ table is obtained. Since the set of information described in these slots is enough to make such a table from the target datatable, our system can draw a graph if a complete frame is obtained.

4.2 "Focal point" frame

The focus of a user's interest that should be emphasized and noticed on a graph is described in a "focal point" frame. This frame consists of three slots: FOC, COMP, and DEG. Each slot takes one slot value. For example, the "focal point" frame for user requirement (ex1) is shown in Figure 3.

The details of each slot are as follows.

- **FOC**
  FOC indicates focal instances mentioned in the user's requirement. For instance, "Kyoto Prefecture" is a focal instance in user requirement (ex1).

- **COMP**
  COMP indicates a comparison object for FOC. For instance, in user requirement (ex1), the comparison object for the FOC (= Kyoto Prefecture) is the other prefectures in the Kinki district.

- **DEG**
  DEG indicates an expression that expresses the degree of FOC to COMP.
  Several expressions such as "increase," "increase slightly," and "increase rapidly" may exist in a declaration-type requirement in order to show the degree of FOC in comparison with COMP. It is necessary to clarify the differences on a drawn graph because the user's intentions are different in these expressions. Therefore, our system decides the emphasis method complying with DEG. For example, if an expression "increase slightly" exists in a user requirement and a line chart is chosen as a suitable graph-type for the requirement, the system draws the chart with a wider range for the Y-axis in order to make the slant of the chart gentler[3].

[3] The current system does not check whether the user requirement is conceptually correct or not.

FOC	COMP	DEG
Y_f = {Kyoto Prefecture}	$\overline{Y \cap Y_f}$	significant

Figure 3. "Focal point" frame of user requirement (ex1)

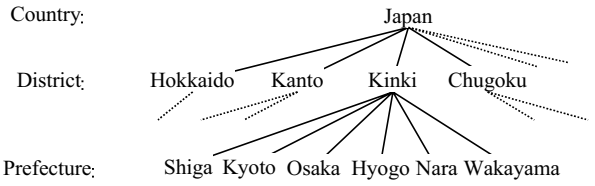

Figure 4. Upper/lower concept knowledge with granularity tags for places

5 ORGANIZING THE NECESSARY DATA

Not all of the tuples in a datatable are necessary to make a graph satisfy a user's requirement. Moreover, a datatable may not be described by an appropriate granularity to make a graph satisfy a user's requirement. Therefore, the system should have the function to select only the necessary tuples from a datatable and aggregate them in a suitable granularity. Our system selects and aggregates tuples from a datatable by the following algorithm.

5.1 Data selection

First, our system selects tuples that satisfy all of the restrictions described as REST(*var*) in the "what to draw" frame from the datatable. For instance, in the case of Figure 1, our system selects tuples that satisfy "measuring date = 1995" and "place $\in$ Kinki district."

As a practical matter, a restriction in the form of upper/lower concept designations is expanded by looking up upper/lower concept knowledge. Figure 4 is an example of upper/lower concept knowledge with granularity tags for places in Japan. For instance, "place $\in$ Kinki district" is expanded to "place = {Shiga, Kyoto, Nara, Osaka, Wakayama, Hyogo}."

5.2 Data aggregation

Next, the system aggregates the selected tuples.

For each *var* in the "what to draw" frame, the system checks the granularity consistency between GRANU(*var*) and the granularity of the corresponding attribute of the tuples. If GRANU(*var*) is coarser than the granularity of the corresponding attribute, the system translates each instance of the tuples into a granularity that is the same as GRANU(*var*) by using operations determined by MEAS(*var*).

Conversely, if GRANU(*var*) is less than the granularity of the attribute, our system lets the user know that the system cannot draw a graph suited to the user's requirement.

After the above operation is finished successfully for every *var*, tuples that have the same instances for every candidate key attribute are aggregated into one tuple as shown in Figure 5.

5.3 Organizing the necessary datatable

Then, the system organizes the datatable based on the subordinate relations between attributes. These subordinate relations can be derived by candidate key attributes in the original datatables. Since $\{A_1, \ldots, A_{n-1}\}$ are candidate key attributes and only A_n is not,

Place	Measuring date	Precipitation
Shiga	1995	7
Shiga	1995	6
Kyoto	1995	13
Kyoto	1995	10

⇓

Place	Measuring date	Precipitation
Shiga	1995	13
Kyoto	1995	23

Figure 5. Example of tuple aggregation

an instance $d_n \in dom(A_n)$ can be determined if all instances $d_i \in dom(A_i)$ $(1 \leq i \leq n-1)$ are designated. A subordinate relation is derived as equation (2).

$$dom(A_1) \times \cdots \times dom(A_{n-1}) \rightarrow dom(A_n). \quad (2)$$

From the above relation, the system reduces the candidate key attributes, each of which has only one instance in the aggregated datatable. By this operation, the system decides on the structure of the organized datatable as equation (3).

$$A_1^* \times \cdots \times A_p^* \rightarrow A_n, \quad (3)$$

where A_i^* is an attribute that has plural instances after the aggregation. For example, in the case of the lower table in Figure 5, the attribute "measuring date" is reduced because the attribute only has the instance "1995".

As the restriction for drawing on a two-dimensional graph, the number of candidate key attributes should be less than two. If more than two candidate key attributes exist after the above operation, the system requires additional restrictions to the user.

For instance, in the case of user requirement (ex1), "precipitation" is a non-key attribute and only "place" has plural instances. Therefore, the following relation is obtained.

$$\text{place} \longrightarrow \text{precipitation} \quad (4)$$

From this relation, the datatable shown in Table 2 is organized.

In the case of user requirement (ex2), "population" is a non-key attribute, whereas candidate key attributes "place" and "gender" have plural instances. Therefore, the following relation is obtained.

$$\text{place} \times \text{gender} \longrightarrow \text{population} \quad (5)$$

From this relation, the datatable shown in Table 3 is organized[4].

6 VISUALIZING DATA

In this process, the system draws a graph suitable for the user's requirement based on the organized datatable.

First, the candidates of appropriate graph-types that effectively express the user's requirement are decided based on the semantic frame and organized datatable.

By analyzing graphs used by newspapers and "white papers," we found that an appropriate graph could be selected by giving attention to five factors[9]: (1) the number of candidate key attributes in an organized datatable, (2) the combination of attributes in an organized datatable, (3) whether a semantic dependency exists between attributes or not, (4) whether a "numerical" attribute is treated as a ratio or not (e.g., whether the sum of the instances becomes 100 or not), and (5) the attribute of FOC in the "focal point" frame.

[4] Our system assumes the closed-world assumption that no rain has fallen anywhere if the target datatable does not have the tuple.

Table 2. Organized datatable from "what to draw" frame in Figure 1

Shiga	Kyoto	Nara	Osaka	Wakayama	Hyogo
1366	1607	1287	1379	1409	1190

Table 3. Organized datatable from "what to draw" frame in Figure 2

	Tokushima	Kagawa	Ehime	Kouchi
man	395000	494000	711000	382000
woman	436000	534000	793000	431000

We construct decision rule sets with a statistical technique based on the above factors to choose appropriate graph-types. By using the rule sets, appropriate graph types can be selected with a probability of 70% or more for all test data, equal to the organized data in this paper.

Then, the system determines the emphasis method and the parts of emphasis by clarifying a significant points in the graph based on the "focal point" frame.

For the organized datatable, tuples having the same instances as the slot value of FOC are chosen as the objects to emphasize and applied with several emphasis operations in the drawing of a graph. For instance, when a bar chart is chosen for the datatable, the graph objects corresponding to the objects to emphasize are displayed in a deep color with a thick edge, whereas comparison objects determined by COMP in the "focal point" frame are displayed in a light color with a thin edge.

Moreover, in the case of a line chart, the appearance of change in the graph becomes intuitively comprehensible by suitably setting the scale of the Y-axis; for instance, to express "the rapid increase," the scale scope is set small to draw a large-change graph, while to express "a slightly increase," the scale scope is set large to draw a small-change graph[8]. This operation depends on the slot value of DEG in the "focal point" frame.

Finally, the system makes a graph using the parameters.

We made a prototype system named KEVIN that uses each of the above processes. Microsoft Excel is used for the process of drawing a graph in our system. The following two sentences are examples of user requirements.

(ex3) *There was much precipitation in Kanagawa Prefecture when the precipitation of Kanagawa Prefecture from January to June in 1995 was compared with that of Tokyo.*

(ex4) *What were the amounts of monthly production of the audio apparatus and the visual apparatus in 1998?*

The output result when user requirement (ex3) is input is shown in Figure 6. The precipitation of both prefectures are shown by plural bar charts. The thickness of the edge and the arrangement of the color are adopted as the emphasis methods. For these effects, a graph that emphasizes the precipitation of "Kanagawa Prefecture in June 1995" is drawn.

The output result when user requirement (ex4) is input to the system is shown in Figure 7. The amounts of monthly production of the two kinds of apparatuses are shown by plural line charts. No emphasis operation is applied because user requirement (ex4) does not have information about a "focal point".

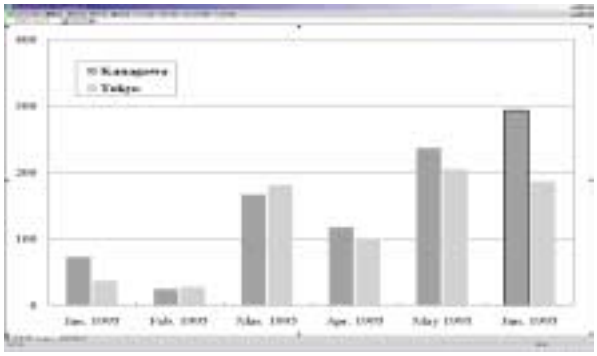

Figure 6. Output example for user requirement (ex3)

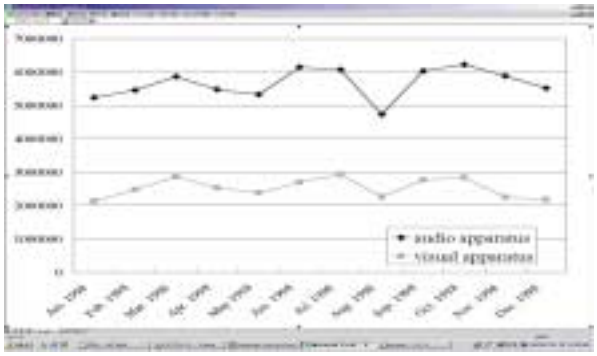

Figure 7. Output example for user requirement (ex4)

7 COMPARISON WITH RELATED WORKS

Information visualization is an effective approach for handling large amounts of data efficiently, and several systems have been proposed[5].

A Presentation Tool (APT) proposed by Mackinlay[4] is one of the major result of the research on visualization. In APT, the system makes a graph by combining graphical primitives, which are the necessary elements for making information visible. The principal objective of Mackinlay is to make a graph language to handle these primitives, and he discusses this based on two criteria: (1) whether the language can express the required information (expressiveness criteria), and (2) whether the language exploits the capabilities of the output media and the human visual system (effectiveness criteria). The discussion of Mackinlay gives guidance on system evaluation in graph generation. However, it lacks a discussion on conversion from a user's requirement to a graph language.

SAGE proposed by Roth[6][3] is a typical system that supports prototyping in an interface for visualizing information. However, it is not enough to provide a method of reflecting the user's requirement, as with APT, because the main point is to select an appropriate graph from the data characteristics available.

The principal target of these visualization tools is to formalize a language by operating data and creating a graph or to choose an appropriate graph style based on the properties and characteristics of data. Therefore, investigating how to reflect a user's requirement is not sufficient.

In addition, the uniformity of the data details is not sufficiently considered, so these systems cannot solve the gaps in details between the suitable data of the user's requirement and the stored data. DBsena[2] and an OLAP (On-line Analytical Processing) based visualization tool[7] have been proposed as frameworks for dealing with details that vary from the user's requirement. These systems, however, do not describe the necessary knowledge and methodologies to automatically visualize information.

Greens' research[1] is remarkable for our interests. Their system makes data visible based on a user's requirement. They proposed a language that represents a user's requirement by using a first-order logic with restricted quantification (RQFOL). Although the language can represent complex user requirements, representation elements such as predicates depend on specific domains. That is, the language does not guarantee applicability to other domains. Moreover, as they are not concerned with data granularity, their system seems incapable of handling various types of data described in different granularities.

In contrast to these systems, our system makes a datatable from a user's requirement and visualizes the data. In other words, the system generates a datatable that reflects the attributes, granularity and restrictions of a user's requirement without being limited to a specific domain. That is, our approach has a lower domain dependency. Moreover, the datatable is organized with a suitable granularity that can be applied to various types of data. Therefore, we conclude that our system has advantages for solving the above problems.

8 CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a new visualization system that automatically draws a graph from a large amount of numerical data corresponding to a user's requirement. By using this system, a user can obtain a graph that effectively reflects his/her required information without considering the necessary parameters for the drawing.

In the future, we will verify the effectiveness of the proposed system by applying data from various domains. We will also develop methods that automatically generate the necessary knowledge from existing databases, LANs, and the Internet.

REFERENCES

[1] N. Green, G. Carenini, S. Kerpedjiev, S. Roth, and J. Moore, 'A media-independent content language for integrated text and graphics generation', in *Proceedings of Content Visualization and Intermedia Representations (CVIR '98)*, pp. 69–75, (1998).

[2] T. Hoshino, M. Tsunakawa, and H. Machihara, 'Dbsena: Information resource management and retrieval engine for multi-database environment', in *IPSJ SIG Notes*, number 2 in 98, pp. 113–120, (1998).

[3] S. Kerpedjiev and S. F. Roth, 'Mapping communicative goals into conceptual tasks to generate graphics in discourse', in *Proceedings of the 2000 ACM Intelligent User Interfaces Conference (IUI2000)*, http://lieber.www.media.mit.edu/people/lieber/IUI/, (2000).

[4] J. D. Mackinlay, 'Automating the design of graphical presentations of relational information', *ACM Transactions on Graphics*, **5**(2), 110–141, (1986).

[5] *Readings in Intelligent User Interface*, eds., M. T. Maybury and W. Wahlster, Morgan Kaufmann Publishers, 1998.

[6] S. F. Roth, J. Kolojejchick, J. Mattis, and J. Goldstein, 'Interactive graphics design using automatic presentation knowledge', in *Proceedings of the Conference on Human Factors in Computing Systems (CHI'94)*, pp. 112–117, (1994).

[7] G. Stumme, 'On-line analytical processing with conceptual information systems', in *Proceedings of the 5th International Conference on Foundations of Data Organization (FODO'98)*, pp. 117–126, (1998).

[8] H. Yonezawa and T. Iida, 'A graphical representation of time series data reflecting an assertion (in japanese)', in *Proceedings of the 12th Annual Conference of JSAI*, pp. 522–523, (1998).

[9] H. Yonezawa, M. Matsushita, and T. Kato, 'Criteria to choose appropriate graph-types', in *Proceedings of the Pacific Rim International Conference on Artificial Intelligence (PRICAI2000)*, to appear, (2000).