データベースシステム 91－8
（1992. 11. 5）

地図メディアのためのアドホック質問のオブジェクト化と視覚化

有川正俊
九州大学工学部情報工学科

データベースでは多目的利用化と高機能化という相反する要求に応じて２種類のオブジェクト（素オブジェクトと加工オブジェクト）を取り扱う必要がある．素オブジェクトは多目的利用を仮定して設計されており，利用する際には加工する必要がある．一方，加工オブジェクトは，特定の利用を仮定して設計されており，現在のハイパーメディアのように利用に依存した視覚的インタフェースの振舞いも定義されている．地図応用では，素オブジェクトと加工オブジェクトはそれぞれ“地理データ”と‘地図データ”に対応する．本論文では，この２種類のオブジェクトをデータベース質問および汎用対話視覚化ツールの観点から統合的に取り扱う枠組みを述べる．

Materialization and Visualization of ad hoc Queries to Integrate Map Databases and Geographic Databases

Masatoshi ARIKAWA

Department of Computer Science and Communication Engineering
Kyushu University,
Hakozaki 6-10-1, Fukuoka 812, Japan
E-mail: arikawa@csce.kyushu-u.ac.jp

Databases must deal with two different types of objects due to the trade-off between the variations and the functionalities. Those are raw objects and processed objects. The raw objects are designed for many uses, and usually processed later for particular uses. On the other hand, processed objects are designed for particular uses, and usually have their own active behaviors. In the case of map applications, examples of the raw objects and the processed objects are geographic data and interactive maps respectively. This paper presents a framework for integrating such two types of objects in the terms of database queries and visualization of both data and queries.

1. Introduction

Computer technology has enabled people to interact with a large amount of data using a large number of functions [Marcus91]. Visual user interface technology has directly contributed to the success. Database researches, however, have overlooked the visual user interface technology [Pilo83, Maier87]. Most of recent commercial database systems on personal workstations provide visual tools to design a database schema, to browse data in it, to make graphs from the retrieved data and so on. Good visual interfaces reduce our efforts required to use and construct databases and provide us with a wide variety of effective views of databases.

Hypermedia (or hypertexts) is considered the most promising paradigm to browse multi-media information, because users can browse it easily by only touching visual buttons on the screen. Moreover, authors can edit it easily because of its simple components. The components of hypermedia are *media* and *visual buttons*. The *media* are units of information for users to obtain at one time, such as texts, images, graphics, animations, videos, voices, sounds, and combinations of them. Media displayed on the screen is called *the current media*. The *visual buttons* are also displayed with current media on the screen. The buttons represent particular media. Users may click one of the buttons on the screen to move from the current media to the media represented by the clicked button.

Hypermedia is a promising approach for map applications as well. For example, car navigation systems require multi-media information, such as images, graphics, and voices. Drivers should be allowed to be able to use multi-media information easily and quickly by means of visual buttons. Most of the current car navigation systems provide maps as image or graphics which are human-made products. All maps used in car navigation systems must be fixed and stored in the form of visual output data. Though some additional information can be overlaid on the prepared maps, the variations of maps are basically limited.

People will soon want their personal maps to be suited to their personal purposes. For example, the scale, the time, and the area of such a personal map should be specified freely by people. People would be willing to select information of their interest from varied geographic databases and to create their own maps from the selected information. It would be impossible to prepare and store all maps, that satisfy everyone's needs, in the fixed form. In this case, it is useful to use ad hoc queries to geographic databases to generate the suitable maps. Selected data through the ad hoc queries should be visualized automatically as *disposable maps*. The disposable maps should provide active behaviors like hypermedia as well as visual information. In other words, the disposable maps should have visual buttons and slides for users to move to the associated maps and details of geographic objects on the maps. We call such maps on computers, which can be manipulated directly by input devices, *interactive maps*.

This paper proposes a framework for users to use ad hoc queries to geographic databases by manipulating visual buttons and visual control devices on the interactive maps without the knowledge of database query languages and geographic database schema. Section 2 describes characteristics of interactive maps and hypermedia from the view point of the object-oriented graphics. Section 3 clarifies the process of creating interactive maps from geographic databases in terms of ad hoc queries. Section 4 introduces "*control planes*" to visualize numeric data and numeric conditions in queries. Section 5 describes a prototype interactive map system being developed. Section 6 gives concluding remarks.

2. Interactive Maps and Hypermedia

2.1 Interactive Maps and the Object-Oriented Graphics

There are various geographic maps for various uses. Most of the techniques developed for producing maps have been based on a static media, *paper*. On the other hand, maps based on a dynamic media, *computer*, are getting popular. We call the maps based on computers *interactive maps*. They provide users with visual environments to move to maps of their interest and to adjust appearances of maps. In the remainder of this paper, we use a word *map* for *interactive map*.

The maps should be based on the technology of the Object-Oriented Graphics [Schmucker86] which enables users to manipulate directly display objects, such as geographic entities on maps, maps themselves, and decorations of maps. The display objects have their own behaviors against users' messages by mouses, buttons, and menus. The correspondences between messages of users and behaviors of display objects are specified by *protocols* in display objects. For instance, double clicks on a school display object on a map cause it to show the detail, a single click on it with the control key down to play the school music, moving it onto a railway station display object to calculate the time needed between them on foot. The protocol in a display object is defined in its *class*. All objects belonging to the same class behave in the same way.

Maps are composed of pieces of display objects as visual components. The display objects can be composed of other display objects as well. Maps are display objects. Windows containing maps are also display objects. In other words, maps are visual components of windows as well as scroll bars, close boxes, and size boxes. Moreover the screen (or desktop) is a display object which has windows as visual components. According to the tree hierarchies, messages of users' mouse clicking are passed from the desktop to the target display objects.

Let us see some examples of operations on a map using Figure 2.1. If we click one of geographic entities on the map, its details of them will appear. The rectangles on the map express the existence of maps of larger scales. If we click one of the rectangles, the corresponding map of a larger scale will appear. If we click the button "NORTH" on the top, we will obtain another map of the same scale which covers the north of the current map. If we click the button "OVERVIEW", we will move to a map of a smaller scale on which the current map occupies a part of it.

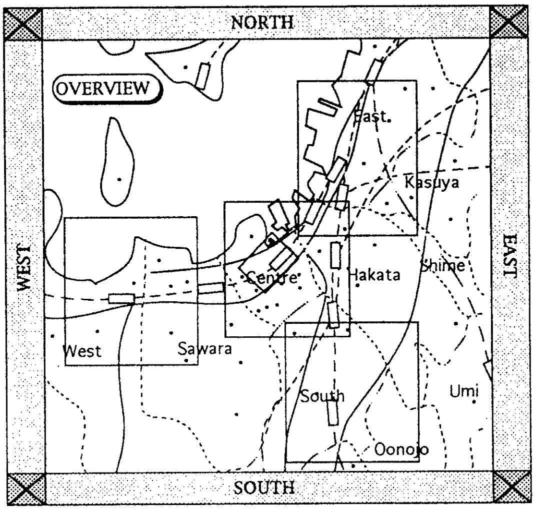


Figure 2.1 An appearance of an interactive map

2.2 Hypermedia

Hypermedia is getting popular as a tool for browsing multi-media information, a tool for presentation and so on. Hypermedia makes good use of object-oriented graphics, and provide users with natural interactive environments.

Current commercial hypermedia is composed of two simple components: *media* (e.g., texts, images, graphics, animations, videos, and sounds) and *visual buttons* (which allow users to move from the current media to the associated media).

Hypermedia is:

(1) easy for users to use;
(2) easy for authors to produce;
(3) fast for users to move to the associated information.

Hypermedia provides users with visual information (expressed in media) and visual buttons on the screen. Users just click a visual button to move to the associate information. Thus users can use hypermedia easily. Simplicity of the components of hypermedia makes authors to master easily how to produce them. Moreover, authors can produce them easily. The quality of hypermedia is up to the scenario created by the authors. It is almost the same as the case of ordinary books.

2.3 Shortcomings of the Current Style of Hypermedia

When information is needed to be updated in the current style of hypermedia, authors must modify it manually in the same way as books. For instance, if a new high way has been made, we should add it to all maps which cover the area. In case of maps, the update can be done automatically using computers and geographic databases. Maps are typical information products which should be and can be automatically updated and composited from databases. On the other hand, news papers and magazines have the opposite characteristics, since they are difficult to produce automatically from databases. If all media in a hypermedia can be automatically generated and updated based on databases, in which most of stored data are made for general and uncertain use, the update problem of hypermedia must be easy.

Hypermedia enables users to *walk* easily through pieces of pre-arranged information by only clicking visual buttons according to the scenario. However, the fixed scenario can never satisfy all purposes of all users. For maps, users may need their own maps which are not prepared in the form of hypermedia. Thus hypermedia cannot provide appropriate information for particular users unless the authors could predict all their uses. Users' choices are limited in hypermedia in a sense. The limitation enables users to use it, while it restricts the flexibility of the information retrieval for users.

For information retrieval systems or relational database systems, users usually use query languages to obtain information of their interest. Query languages provide users with the flexibility of the information retrieval, but it is not easy for users to use them compared to the visual user interface of hypermedia.

Characteristics of ad hoc queries should be introduced into the current hypermedia to allow users to have more freedom in retrieving information. Visual interface to use ad hoc queries should be in the same way of that of hypermedia. For maps, users should be able to obtain new maps which are results of their ad hoc queries to geographic databases. Maps based on geographic databases are specified by users' ad hoc queries. Section 3 describes a framework of using ad hoc queries of maps as traversing instant links between maps in hypermedia.

3. Interactive Maps Based on Geographic Databases

3.1 Conceptual Objects and Display Objects

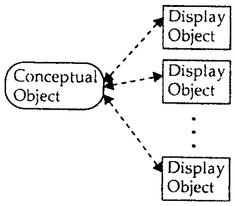
Conceptual objects are stored as entries in databases and are designed for many purposes. On the other hand, *display objects* are results of visualizing the conceptual objects by certain visual methods. Multiple display objects can refer to the same conceptual objects (Figure 3.1 (a)). A map consists of multiple display objects (Figure 3.1 (b)). If conceptual objects are updated, the changes would appear on the maps containing the display objects corresponding to the updated conceptual objects.

3.2 Interactive Maps as Snapshots of Results of Queries to Geographic Databases

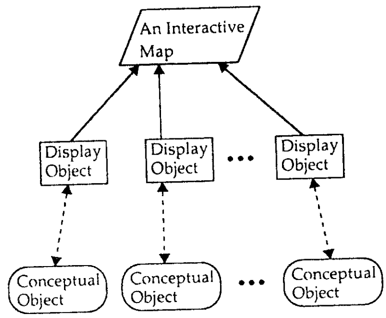
The process of making maps from geographic databases is as follows (Figure 3.2):

(1) retrieve conceptual objects of users' interest from geographic databases by ad hoc queries;

(2) select the methods to visualize the retrieved conceptual objects as display objects in a map.



(a)



(b)

Figure 3.1 (a) Conceptual objects referred by multiple display objects; (b) An interactive map composed of multiple display objects

A set of conceptual objects selected from databases is called *a database view*. The database view (*DV*) is denoted as a pair of *C* and *D*, that is <*C*, *D*>. *C* represents a set of conditions in an ad hoc query. *D* represents a geographic database such as a school database, a river database, and an administration area database. *A visual layer* (*VL*) which corresponds to a *DV* is denoted as a pair of the *DV* and a visual method *VM*, that is <*DV*, *VM*>. The visual method *VM* specifies characteristics of display objects such as the pattern, the color, and the behavior against a single click. *A map* (*IM*) is defined as a set of *VL*. A map is a snapshot of the result of a query to geographic databases. The map itself can be a persistent object, and be stored in *a map database*. Unless queries of a map are changed, the map can be treated as a piece of media in hypermedia.

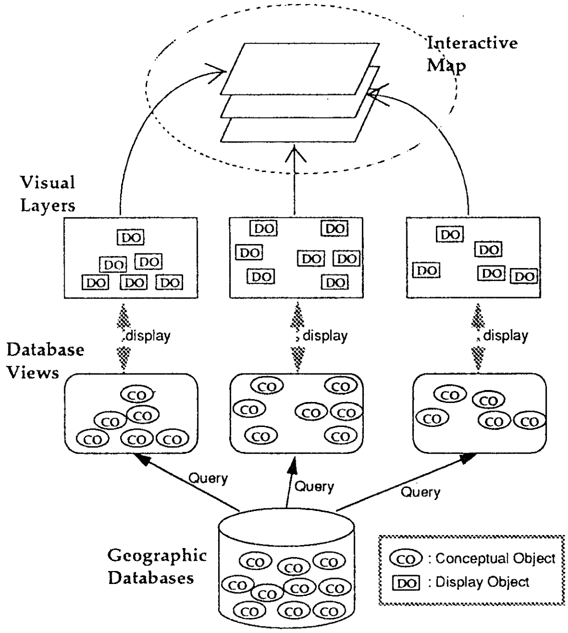


Figure 3.2 The process of creating a map from geographic databases

There are some common conditions in queries of database views, such as the area and the time. The common conditions (*CC*) are introduced to a map *IM* as an additional component. The *CC* is a set conditions which are common for conditions of queries to every database. The common conditions *CC* of a map *IM* are parts of *C* of every *DV* which belongs to the *IM*. For example, if the *time* condition of the *CC* of a map is changed from 1991 to 1990, all *DV*'s of the maps are updated by the queries in which the time condition is 1990. If we add information of another new database *NDB* to the map, the database view of the *NDB* is automatically created according to the common condition "*time* = 1990".

3.3 Differential Changes of Queries as Traverses between Virtual Media

This section shows the following three modifications of maps.

(1) changes of visual methods of a visual layer;
(2) changes of queries of database views;
(3) additions/deletions of database views.

Changes of visual methods of visual layers cause changes of characteristics of the display objects. For instance, we change the color of all display objects on a certain visual layer.

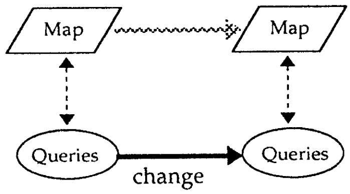


Figure 3.3 Moving to the next map by changing the current queries

It is natural to assume that a map is a visualization of the queries. Usually, users make their maps incrementally by making the corresponding queries incrementally (Figure 3.3). Users may decide the next changes in the current queries after seeing a visualization of them, that is the current map. For instance, when we want to zoom in a map, the value of the common condition "*Area*" of the current queries should be made smaller. When we want to see the information of railways on the current map which is not an entry on it, a query to a railway database should be added to the current queries. Resultant queries to the railway database must include the common conditions (*CC*) of the map.

For maps based on the current style of hypermedia, obtaining the next map means traversing prepared links from the current map to the associated map. For maps based on geographic databases, changes in queries for the current maps can be considered traversing a virtual link from the current map to virtual maps which are composed instantly as a result of ad hoc queries. Section 4 introduces *control planes* which provide users with an environment to visualize conceptual objects and conditions of queries on a map and to use them in the similar manner of hypermedia.

4. Control Planes

4.1 A Coordinate Plane and Value Objects

Unless we map both conceptual objects and conditions of queries on a certain coordinate, we cannot see and manipulate them directly. When we visualize conceptual objects in the forms of display objects, the characteristics and the coordinates of the display objects on the screen should be determined by the attributes of the conceptual object to project and the scope of the coordinates. This paper proposes a type of display objects called "*control planes*" to specify the coordinates for placing display objects which are visualizations of both conceptual objects and conditions of queries. The control plane is composed of three kinds of display objects, *a coordinate plane*, *value objects*, *decoration objects*. The scope of a *coordinate plane* defines the extent of a certain coordinate system to display geometric information. The *value objects* specify geometric regions contained in the scope of the coordinate plane. In most cases, the value objects are the results of visualizing some of attributes of conceptual objects. The *decoration objects* control the scope of the coordinate plane.

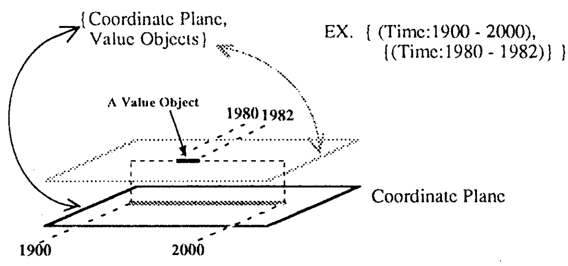
Control planes are introduced to integrate visual control parts, such as buttons and slides, and visualization of numeric attributes of conceptual objects. We will show examples of control planes. Figure 4.1 (a) is an example of one-dimensional control plane. The scope of the coordinate plane is the duration from 1900 to 2000. A value object occupies a region in the scope, which corresponds to the duration from 1980 to 1982. For example, its region may present the time when the president of a university was a woman. The region can be composed of separated regions. We can project multiple value objects on the same coordinate plane. The visual result can be considered a one-dimensional map. If we modify the region of the value object by visual manipulations, the visual change affects the attribute values of the conceptual object which corresponds to the value object. Control planes can be used to visualize conditions of queries to databases in addition to visualization of conceptual objects. For example, the duration from 1980 to 1982 can be used as a condition to select universities through the *time* attribute indicating when they were established. In this case, changes of the region means changes of the query. If we move the region from 1900 to 2000 at a certain speed by a mouse, universities would appear and disappear. Control planes can be considered as generalized *visual slides* (or scroll bars) in case of projections on one-dimensional planes. Value objects may correspond to nobs on the visual slides.

Figure 4.1(b) is an example of two-dimensional control planes, which are simple extensions of one-dimensional ones. Supposing that a black dot in the middle layer expresses the location of an university. We can then project more universities on the same coordinate plane. As a result, the control plane serves as a university map. The scope of the coordinate plane specifies the area which users want to see.

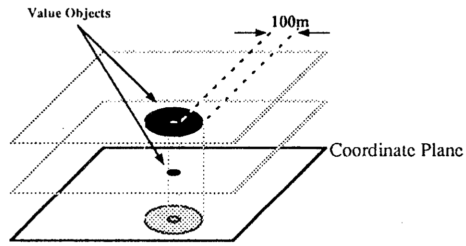
We can derive new value objects from existing value objects on a map. A black donut region is the area where the distance from the black dot is less than 100 meters. The black donut can be used for various purposes. For example, we may use it as a condition of a query to select female colleges which is located in less than 100 meters from the university. If the size of the black donut region is associated to the distance between the black dot and the position of the mouse cursor, we can change the condition of the query dynamically. Moreover if there are many universities on a map, we can create many donut regions of them simultaneously by moving mouses. The resultant query is specified to select female colleges in less than 100 meters from more than one of all universities on a map. Usually, we specify the geometric conditions as rectangles because they are defined easily and processed efficiently.

4.2 Decoration Objects

Usually, scopes of coordinate planes can be used as conditions of queries. Changes of the scopes can induce changes of the conditions. We introduce *decoration objects* to control the scope. We explain decoration objects using an



(a) An One-Dimensional Control Plane



(b) A Two-Dimensional Control Plane

Figure 4.1 Examples of control planes

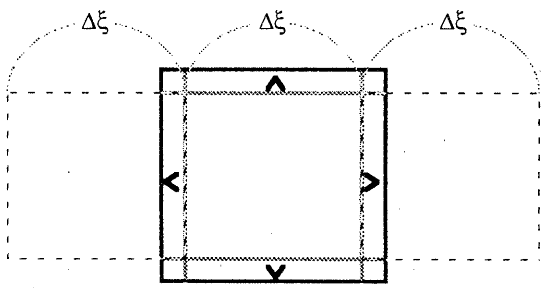


Figure 4.2 Decoration objects to control the scope of the coordinate plane

example of Figure 4.2. There are four decoration objects in the four ends of the control plane. Clicking the right one may shift the scope of it toward the right direction by $\Delta\xi$. Others behave in the similar way. In case of maps, after users' clicking the right decoration object, they obtain a derived map which covers the east of the current map. The facilities of decoration objects let us use queries unconsciously. As a result, we can use maps based on ad hoc queries in the same way of hypermedia at the point of browsing information by only clicking buttons.

5. A Prototype Interactive Map System

This section describes a prototype interactive map system being developed on a Macintosh using Think C, which is almost C++, and Think C Class Library [Schmucker86, ThinkC89]. The characteristics of the prototype are as follows:

(1) all maps are created and updated based on geographic databases;
(2) users can create maps of arbitrary regions, scales, and geographic information;
(3) all components on maps are display objects which can respond to messages from users;
(4) users can derive new maps from existing maps by changing the corresponding queries of the existing maps;
(5) users can update contents of our geographic database and conditions of corresponding queries of maps by manipulating display objects on maps.

The rest of this section explains the prototype system. The map in Figure 5.1(a) includes coastlines, universities, and colleges in the east part of Fukuoka City. The queries of the map specify that all databases to be selected are those of coastlines, universities, and colleges, and that they are located in the east part of the city. Selections of databases are specified by menus which present available databases. Black dots are display objects of universities and collages. Polylines are display objects of coastlines. If we doubleclick a black dot or a coastline, the detail will appear on the screen. Appearances of display objects can be adjusted by menus. Figure 5.1(b) is the example. In the map, universities and collages are presented as pairs of a black dot and a Japanese name which has the properties of characters, such as "its font = Osaka" and "its size = 9 points".

Let us derive a new map from the current map with the title "Interactive Map 1" in Figure 5.1(a). We create a rectangle display object, which expresses the area to be set in the condition of queries of the new map, on the current map to zoom in. After that, the corresponding new map is derived. The new map is behind the current map in Figure 5.1(a). A doubleclick on the rectangle in "Interactive Map 1" causes that the new map floats up (Figure 5.1(b)). The new map is named "Interactive Map 2". The display object which has the Japanese phrase "元の地図へ", which means "to the source map", allows us to move to the source map "Interactive Map 1" by clicking it.

In this way, the prototype system provides end-users with direct links to move between existing maps as well as an infinite number of virtual links which can be set by creating rectangles on existing maps as common conditions of ad hoc queries. Collections of existing maps can be used as stacks of hypermedia. Traversing virtual links means using unprepared stacks which are created instantly.

In Figure 5.1(c), the window size of "Interactive Map 2" is made smaller. The rectangle on "Interactive Map 1" is a representative of "Interactive Map 2". It shows us the area of "Interactive Map 2". If we drag the rectangle into another place, the area of "Interactive Map 2" will change. In other words, the corresponding queries are executed again, and a new map named "Interactive Map 2" for another area is created. Resizing the rectangle also updates the corresponding map. If we dispose "Interactive Map 2", the corresponding rectangle on "Interactive Map 1" will disappear. The display object "元の地図へ" is a representative of "Interactive Map 1". Disposing "Interactive Map 1" causes to dispose the display object "元の地図へ". Moreover If we click an arrow button beside a scroll bar on "Interactive Map 2", the map shifts to that direction. As a result, the corresponding rectangle on "Interactive Map 1" also shifts to the specified direction. Figure 5.1(d) shows derived maps on which additional geographic information is overlaid.

Maps are plates to project not only geographic information, but also conditions of the corresponding queries (Figure 5.2). Figure

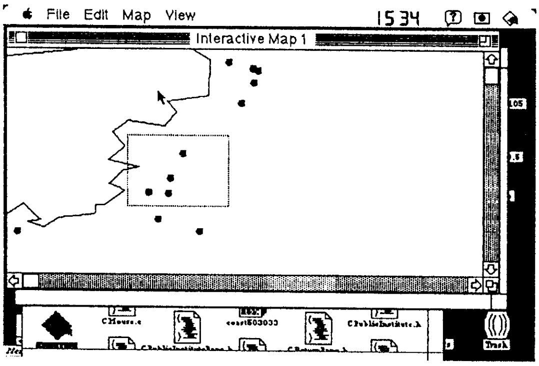


Figure 5.1(a) A map of the east part of Fukuoka City

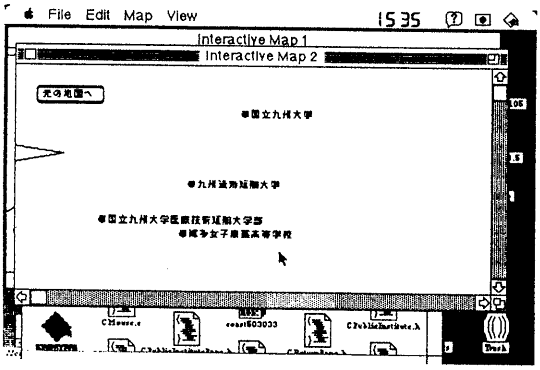


Figure 5.1(b) A map as the result to zoom in the map in Figure 5.1(a)

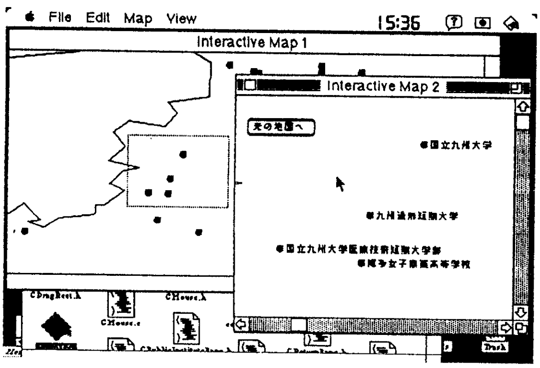


Figure 5.1(c) The source map and the derived map

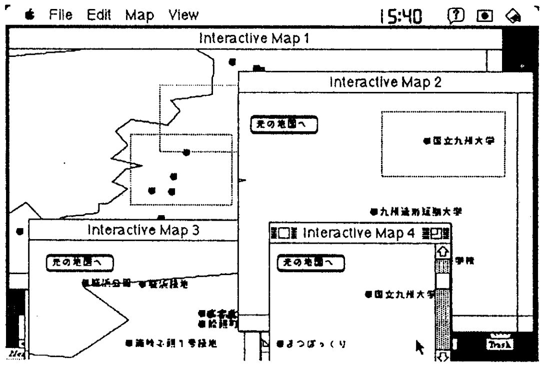


Figure 5.1(d) Derived maps with additional informations

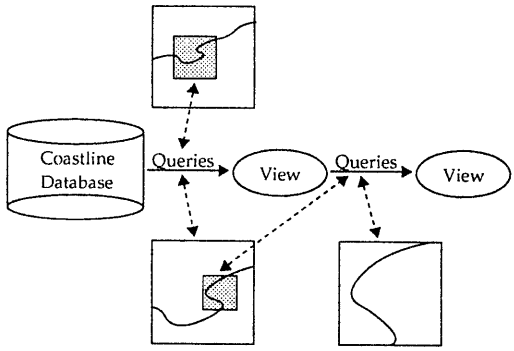


Figure 5.2 Visualization of ***Area*** conditions of ad hoc queries

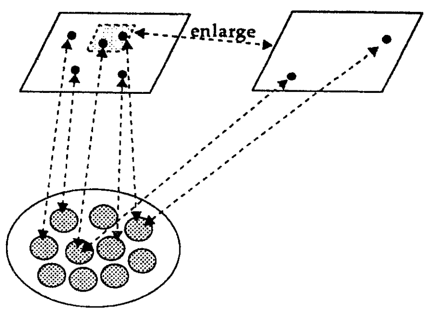


Figure 5.3 References between conceptual objects and display objects

5.3 illustrates references between conceptual objects and display objects. Maps can also be stored in databases. They can also be conceptual objects which are projected on other maps in the form of rectangle display objects. As a result, we can update conceptual objects by manipulating the corresponding display objects directly.

6. Concluding Remarks

Databases must deal with two different types of objects due to the trade-off between the variations and the functionalities. Those are raw objects and processed objects. The raw objects are designed for many uses, and usually processed later for particular uses. On the other hand, processed objects are designed for particular uses, and usually have their own active behaviors. In case of map applications, examples of the raw objects and the processed objects are geographic data and interactive maps respectively. This paper presents a framework for integrating such two types of objects in the terms of database queries and visualization of both data and queries. The framework enables us to use both geographic databases and map databases in the same style of visual user interfaces as hypermedia.

Tanaka also takes a similar approach using queries to create instant links for hypertexts [Tanaka90]. In his approach, queries are used to select two sets of objects to connect directed links between all elements of them. The links come out from objects in one set, and get to those in the other. On the other hand, the approach in this paper focuses on changes in the current queries which express the current map on the screen. Changes in the current queries were considered traversing virtual links. As a result, clicking can use both ordinary fixed links of hypermedia and virtual links based on ad hoc queries in the same style of visual user interfaces.

Acknowledgement

I would like to thank Prof. Akifumi MAKINOUCHI (Kyushu Univ.) and Prof. Yahiko KAMBAYASHI (Kyoto Univ.) for their helps and valuable comments. Prof. Yutaka TSUTSUMI (Ariake National College of Technology) often joined our seminar to discuss the implementation of our interactive map system. I acknowledge his good advice. I thank Dr. Hirofumi AMANO (Kyushu Univ.) for his good comments on the paper. The digitized Geographic Information is supplied by the Geographical Survey Institute of Japan.

References

[Ari91] Arikawa, M., "A View Environment to Reuse Class Hierarchies in an Object-Oriented Database System", Proceedings of the Second International Symposium on Database Systems for Advanced Applications,1991, pp. 259-268.

[Barr86] Barr, R., "Automated Cartography and Geographical Information Systems", in *Advances in Computer Graphics II*, Springer-Verlag, 1986, pp. 29-53.

[Mackinlay86] Mackinlay, J., "Automating the Design of Graphical Presentations of Relational Information", ACM Transactions on Graphics, Vol. 5, No. 2, 1986, pp. 110-141.

[Maier87] Maier, D., Nordquist, P., and Grossman, M., "Displaying Database Objects", Expert Database Systems, 1987.

[Marcus91] Macrcus, A. and Dom, A. v., "User-Interface Developments for the Nineties", IEEE Computer, Vol. 24, No. 9, 1991, pp. 49-57.

[Pilo83] Pilote, M., " Data Modeling Approach to Simplify the Design of User Interfaces", Proceedings 9th VLDB, 1983.

[Schmucker86] Schmucker, K. J., "Object-Oriented Programming for the Macintosh", Hayden Book Company, 1986.

[Tanaka90] Tanaka, K. and Nishikawa N., "TextLink-II: An Object-Oriented Hypertext-Base System," Proceedings of the Far-East Workshop on Future Database Systems, 1990, pp. 26 - 37.

[ThinkC89] Think C User's Manual, Symantec Corporation, 1989.