Dynamic Map Synthesis Utilizing Extended Thesauruses and Reuse of Query Generation Process

Ken'ichi Horikawa†, Masatoshi Arikawa‡,
Hiroki Takakura* and Yahiko Kambayashi*

†Sumitomo Electric Industries, ‡Hiroshima City University, *Kyoto University

Abstract

In this paper, we will discuss a map synthesis system which handles static information (geographic objects) as well as dynamic information (traffic conditions, weather, etc.). The system is considered to be general purpose and can generate arbitrary maps according to the user specification. It is very difficult for a user to specify a exact query which corresponds to the required map. Furthermore even selected objects are appropriate, the map may not be proper to be displayed. The system automatically generates missing information or find errors in the user specification. The result is feedback-ed to the user and further refinement is made. For the purpose we use geographic domain thesaurus which contains aggregation and other semantic relationships as well as conventional thesaurus hierarchy. Especially compatibility levels of objects play important roles to generate maps and find errors in the specification. We will also discuss methods to reuse similar map generation processes.

1 Introduction

With recent advances of computer technology, computer systems which can deal with maps effectively have been developed. Personal map systems, such as car navigations, have been very popular in the past few years. In the near future, it will be important to integrate geographic information such as rivers, mountains and boundaries between cities, and geographically referenced information such as weather, timetables of trains, traffic jams and advertisements of shops. The market for the geographic information services is considered to be rapidly growing[2, 3].

The current geographic information services are mainly based on the conventional hypermedia systems[4, 5]. The conventional hypermedia systems usually consist of two components: nodes which are pieces of ready-made maps and links which correspond to executions of traversing among the maps. Users cannot easily obtain any scale of the maps in combination with the geographical information. Database systems keep all objects shared by users and only required objects are retrieved for generating view. This paper applies the concept of the database systems to map generation. Our system can generate more suitable maps dynamically for a specific purpose (Figure 1). This paper calls this **dynamic information synthesis.**

Generally, a user describes specifications of "Theme", not those of **"Background"**. If maps show only thematic parts, spatial relationships are not clearly displayed. However, the user may feel difficulties to specify background parts.

This paper discusses **"Query Expansion"**, a framework to translate user's query into another query as shown in Figure 2. Semantic relationships between terms in geographic domain are stored in **"geographic domain thesaurus"**. Geographic domain hierarchical levels (GDHLs) which relate spatial expanse are also useful for query processing. **Query databases** store both general knowledge to generate maps in advance and users' queries to re-utilize prepared for the next queries.

GIS 97 LasVegas Nevada USA
Copyright 1997 ACM 1-58113-017-1/97/11..$3.50

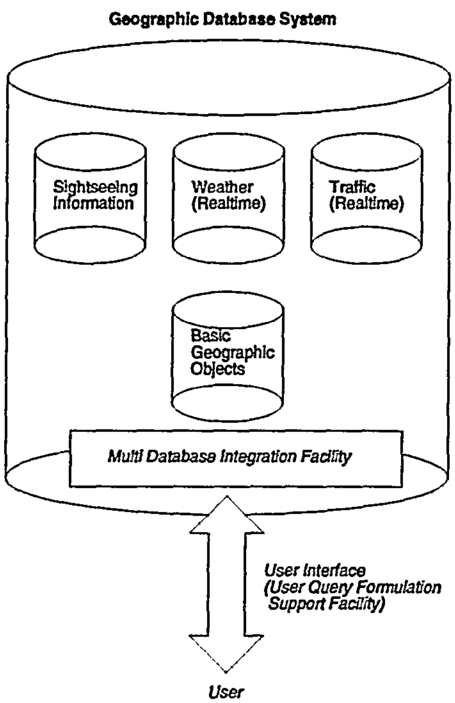

Figure 1: Overview of a Geographic Database System

2 Outline of a Map Synthesis Procedure

2.1 Overview

This subsection describes a framework in which users' initial queries are converted to more feasible and logical descriptions to retrieve useful maps. A query formulation procedure consists of the following steps.

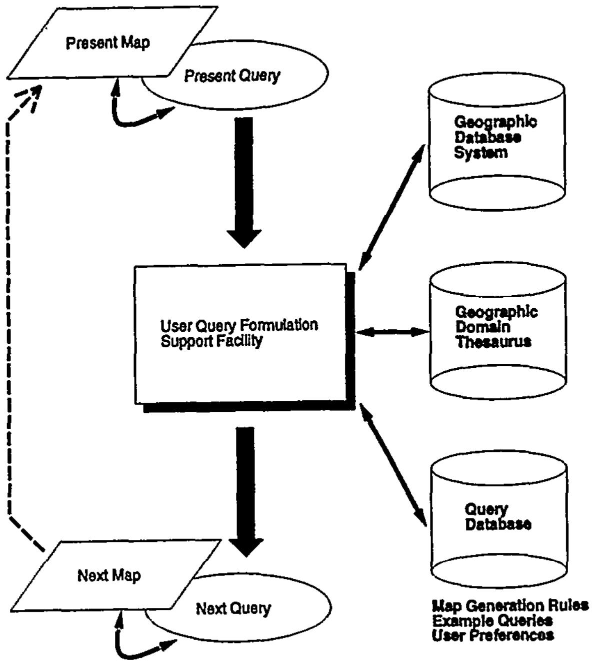

Figure 2: Query Formulation Processes

1) **Inducing users' queries:** Without detailed knowledge of contents of systems, users may be troubled to make their queries. To induce the queries, contents of a database and examples of queries should be easily understood by users. (Section 4 and 5)

2) **Resolving ambiguity of users' queries:** Without detailed knowledge of systems, users may formulate ambiguous queries. Such ambiguities should be resolved by interaction between users and systems. If an query existing in the database is found to be similar to a current query, clarification steps of similar query may be applied to current one. (Section 3 and 4)

3) **Correction of users' queries:** The users often formulate incorrect queries. Methodologies for correcting errors have been studied for a long time, so this paper does not discuss them in detail.

4) **Completion of users' queries:** Users usually write queries by specifying only their thematic parts which the users concern, e.g., museum, park, station. It is troublesome for users to specify background parts of maps, e.g, load, river, landmark. Such incompleteness of users' queries should be compensated by utilization of knowledge-bases. (Section 2.2 and 5.1)

5. **Application of users' preferences:** Users have different view for maps, because they have difference in classification of geographic objects, in familiarity of target place and in preferences for visualization. In this paper, these three differences are called "preferences".

To realize these functions, knowledge of cartography and former queries are stored in a query database. Geographic domain hierarchical level (GDHL) is utilized as criterion to decide if cartographic rules and former queries are applied to current queries.

2.2 Cooperative Query Formulation

Queries which users usually describe are called logical descriptions of maps. The descriptions require large number of specifications. In this section, the following simple example of the descriptions is used in order to simplify a basic ideas of query formulation support processes.

(Area:{...}, Theme: {...}, Background:{...})

When a user specifies only "schools" for the **Theme**, the logical description becomes,

(Area:∅,Theme:{schools}, Background:∅)

If the description is executed, retrieval from the geographic databases results in selection of all school entries, so that a map cannot visualize the results. The map system considers that the results is due to inadequate users' specification, and asks the users to specify **Area**. The map system provides the users with some geographic domain thesauruses as menu style interfaces. The users can select well-used areas associated to their intention. If the user selects "Kyoto City" for the **Area**, the logical description becomes as follows.

(Area:{Kyoto City}, Theme:{schools}, Background:∅)

The map system would infer some combinations of geographic features for the **Background** in consideration of the **Area**'s scale and the **Theme**'s contents. The inferring background may results in like this.

(Area:{Kyoto City}, Theme:{schools}, Background: {city border lines, wards border lines, highways, national roads, prefectural roads, rivers}**)**

After these process, the naive logical description is decided, but visualization methods for the logical description have not been decided yet. The examples of visualization are graphic properties of geographic objects, such as colors, fonts and sizes, and are combinations of attributes to be displayed on a screen. The visualization is discussed in Section 6.

3 Geographic Domain Thesaurus

This section defines **compatibility levels** between classes (or concepts) in the **geographic domain thesauruses.** The similarity between logical descriptions is convenient for users to formulate queries for maps. If the existing description which is similar to user's description has visualization methods for retrieved results, the visualization methods can be applied to user's description. The compatibility levels are utilized to estimate similarity between logical descriptions.

[Geographic Domain Thesauruses]
The geographic domain thesaurus are acyclic graphs (C, L). C and L are denoted a set of **classes** and a set of **directed links** respectively. An Example of the geographic domain thesaurus is illustrated in Figure 3.
Here, the following two notations are defined.

$lev(c_1, c_2)$: the minimum number of links, i.e., the shortest path, between classes c_1 and c_2.

$lub(c_1, c_2)$: a set of classes which are the least upper bounds of classes c_1 and c_2. The least upper bounds of two nodes (or classes) in a tree mean the nearest common ancestors.

[Compatibility Level between two classes]
It is defined as follows,

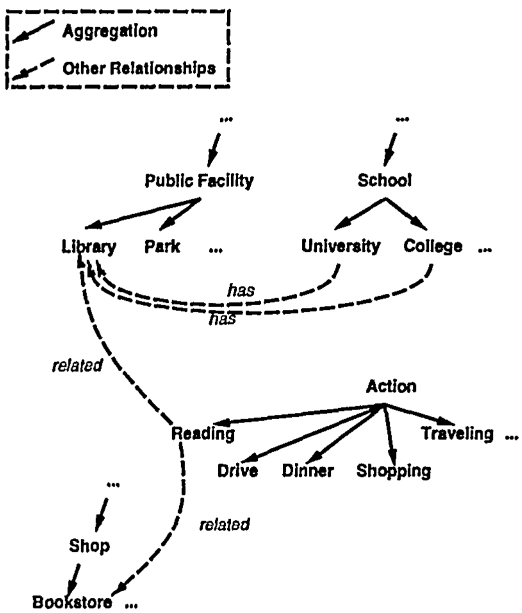

Figure 3: Geographic Domain Thesauruses

$$cmpa(c_1, c_2) = min(min_{c \in lub(c_1,c_2)} max(lev(c_1, c), lev(c_2, c)), hasSameParent(c_1, c_2))$$

Here,

$$hasSameParent(c_1, c_2) = \begin{cases} 0 & \text{if there is the same parent for both } c_1 \text{ and } c_2 \\ \infty & \text{otherwise} \end{cases}$$

"$cmpa(c_1, c_2) = 0$" means that these two classes can be compatible and it is considered appropriate that visualization methods defined in one class can be applied to another.

[Compatibility Level between two sets of classes]
The compatibility level is defied as follows.

$$cmpa(SC_1, SC_2) = \sum_{c_1 \in SC_1} \{(cmpa(c_1, SC_2) : 1)\} + \sum_{c_2 \in SC_2} \{(cmpa(c_2, SC_1) : 1)\}$$

Here,

$$cmpa(c, SC) = min_{c' \in SC} cmpa(c, c')$$

$$\{\ldots, (x_p : a_p), \ldots\} + \{\ldots, (y_q : b_q), \ldots\} = \{\ldots, (z_r : c_r), \ldots\}$$

$$c_r = \begin{cases} a_p + b_q & \text{if } (\exists p, q)(z_r = x_p = y_q) \\ a_p & \text{if } (\exists p)(z_r = x_p) \wedge (\forall q)(z_r \neq y_q) \\ b_q & \text{if } (\exists q)(z_r = y_q) \wedge (\forall p)(z_r \neq x_p) \\ 0 & \text{if } (\forall p)(z_r \neq x_p) \wedge (\forall q)(z_r \neq y_q) \end{cases}$$

The compatibility between the sets becomes higher if the number n of $(0{:}n)$ is larger and becomes lower if the number n of $(\infty{:}n)$ is smaller. Using this rule, we can find the most compatible sets of classes among existing sets of classes.

[Compatibility Level between two areas]
Compatibility level between two areas is defined by the similar way as classes.

$$cmpa(area_1, area_2) = cmpa(SC(area_1), SC(area_2))$$

Table 1: An Example of GDHL

Level	*Area*(x)	*Distribution*(x)
8	$(500km)^2 \sim$	Country
7	$(200km)^2 \sim$	Region
6	$(70km)^2 \sim$	Prefecture
5	$(10km)^2 \sim$	City
4	$(3km)^2 \sim$	Ward
3	$(0.5km)^2 \sim$	Town
2	$(0.1km)^2 \sim$	Section
1	$(0km)^2 \sim$	Corner

Here,

$SC(area)$: the set of classes which corresponds to *area*.

[Compatibility Level between two maps]
Compatibility level between two maps can be derived by the similar formulation as sets of classes.

$$cmpa(map_1, map_2) = (cmpa(area(map_1), area(map_2)) : 1) + cmpa(SC(map_1), SC(map_2))$$

Here,

$area(map)$: area which is a part of *map*.

$SC(map)$: a set of classes which are specified by the logical descriptions of *map*.

4 Detection of Incorrect Combinations of Words

4.1 Geographic Domain Hierarchical Levels

There are some incorrect combinations of words as ad hoc queries for maps generation. The following is an example of incorrect and ambiguous ad hoc queries.

[EX.] "Bookstores near Kyoto City"
The scales of the two entities, "bookstores" and "Kyoto City", are mismatched. If the ad hoc query is executed as it is, the retrieval results in access to a set of too many bookstores along the border line of Kyoto City except located within Kyoto City. Usually, the map with too many objects is not useful, even if it may match users' intention.

We introduce Geographic Domain Hierarchical Levels (GDHLs) to find and solve the incorrect combinations of words and the semantic ambiguity of words. Table 1 shows an example of GDHL. All geographic objects have their physical and logical scales. The physical scale is called *Area*, denoted by *Area*(x) where x is a geographic entity in GDHL. On the other hand, the logical scales are defined from the viewpoints of distributions of geographic objects. The *Distribution*, denoted by *Distribution*(x), is considered the logical scale of the x.

We show some solutions for the above example using GDHL.

(B1) *Area*(Bookstores) $\simeq$ Level:1
(B2) *Distribution*(Bookstores) $\simeq$ Level:2
(KC1) *Area*(Kyoto City) $\simeq$ Level:5
(KC2) *Distribution*(Kyoto City) $\simeq$ Level:8
(Since Kyoto is famous all over the world.)

The closest combination is the pair of (B2) and (KC1) with the distance 3. As the distance 3 is long, the combination of "Bookstores" and "Kyoto City" may be considered

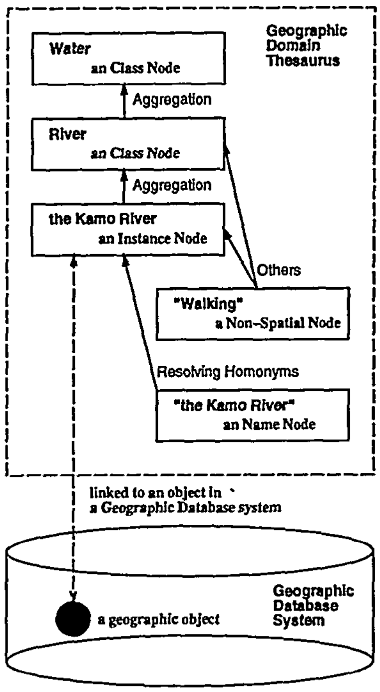

Figure 4: Nodes and Relationships in Geographic Domain Thesaurus

inappropriate. Users should be suggested to change their ad hoc queries into another better pair with the short distance.

4.2 View Functions of Geographic Domain Thesaurus as a Database

As described, geographic domain thesaurus and GDHL support for users to formulate queries. Figure 4 is an example which shows nodes and relationships in a geographic domain thesaurus.

Geographic domain thesaurus can be regarded as a geographic database. Therefore view functions of the geographic domain thesaurus are important functions because view functions of databases enable data set to be transformed into more useful information.

Detailed model of the geographic domain thesaurus is extended to classify nodes into four types.

Instance Node corresponds to a set of geographic objects in geographic databases, e.g., "John's house".

Class Node represents categories to classify geographic objects, e.g., "houses".

Name Node has links to instance/class nodes in order to resolve ambiguity of names which homonyms cause.

Non-Spatial Node, like "Shopping", is introduced to make geographic domain thesaurus more informative.

For view functions, relationships among nodes have the following semantic types.

Resolving Homonyms : Relationships among a name node and instance/class nodes.

Aggregation : If a node consists of several nodes, relationship among these nodes are called "aggregation". "Aggregation" is a one-to-many relationship.

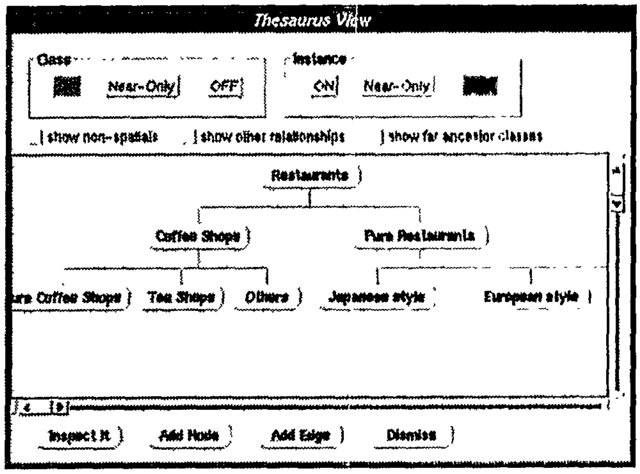

Figure 5: View Function of Geographic Domain Thesaurus

Others : Relationships among a non-spatial node and instance/class nodes.

Figure 5 shows an example of a geographic domain thesaurus and their modification facilities. Users can generate queries in an easy form for a geographic domain thesaurus to browse schema and contents of database, and can modify the queries in order to make suitable for users' properties. The window shows information on restaurants. Users can navigate thesaurus and learn about the database.

Geographic domain thesaurus can integrate GDHL. Each node has one more attributes for storing GDHL. For example, instance nodes can inherit a GDHL from its direct ancestor class node. However such nodes as non-spatial nodes cannot relate a geographic domain hierarchy.

5 Reuse of Synthesis Process

This section discusses another database, i.e., a query database, which the query expansion method utilizes. The database stores queries prepared at setup time and those generated by users. Users apply these queries to new query generation process. This application is called "reuse".

5.1 Basic Flow of Query Expansion by Query Database

This section describes basic flows of user-generated ad hoc queries.

[A. Generating skeletons for logical descriptions of a naive map]

When the map system finds that a user gives logical descriptions does not contain sufficient components, the system utilizes GDHL and suggests the user for selecting candidates as discussed in Section 2.2.

[B. Offering appropriate logical descriptions]

After making the skeleton, the map system searches compatible example of logical descriptions existing in the query database and offers the user that the compatible logical descriptions can be inferred to produce more efficient maps.

[C. Replacing each components of logical descriptions]

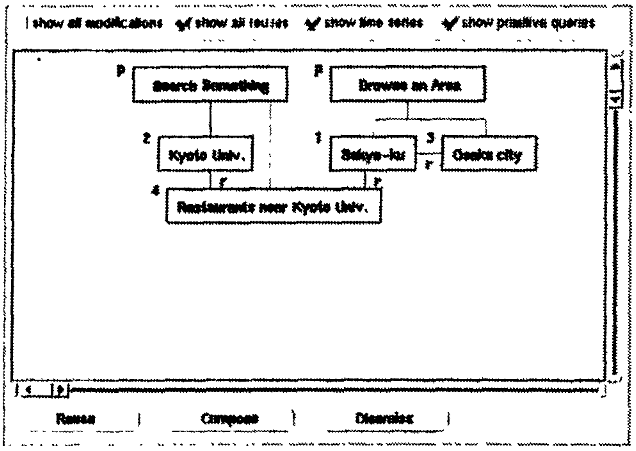

Figure 6: View Function of Query Database

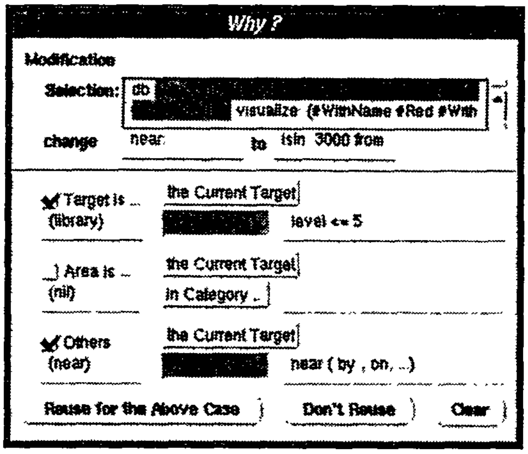

Figure 7: Description of Modification Reasons

For each component of the logical descriptions, the map system shows the user some candidates for replacing it according to the compatibility levels of components.

[D. Arbitrary construction of logical descriptions without compatible levels]
The user is also allowed to utilize the geographic domain thesauruses and personally defined thesauruses through the corresponding menu style interfaces in order to construct logical descriptions without the candidates inferred.

5.2 View Functions of Query Database

Extension of the user interface is very efficient in order to utilize view functions of the query database. In the query database, queries are stored not in a set, but in a graph. Each node of the graph is a corresponding query. Edges among query nodes, which correspond to users' modification of queries, are classified into two categories.

Change of request : A user may change his/her request while browsing a database. This type of modification divides a query graph into several groups. In this case, any relationship may not exists between the original request and modified one.

Modification for adjustment : Usually the user modifies a little part of the request, i.e., adjust the description. Since this type of modifications may be reused for another queries, description of reason for modifications is effective. Several modifications can be merged as one operation.

Here "Thesaurus Relationship" is introduced to reinforce view functions of query databases.

Thesaurus Relationship : Queries in a query database are classified into groups, by cutting "modification of request" edge. "Thesaurus Relationship" group is a part of these groups under a certain criterion. Geographic domain thesaurus can be used as this criterion.

Figure 6 shows a simple example of a view from a query database. Such an interface is considered to have the following advantages.

- The interface informs users about contents of databases and feasible queries. In other words, users can learn how to write queries by referring to existing queries.
- It leads users' requests to almost equivalent queries by referring to existing queries.

5.3 Utilization of Modification Reasons

This subsection shows another method of a facility in order to support formulation of user query. This method utilizes reasons why users modified queries. The reasons are stored in the query database and searched as the reuse conditions that users modify their queries. The query database keeps each modification and its reuse, so that the database becomes smarter during supporting users' query generation.

Figure 7 shows an example of reason description. On this window users can distinguish modifications of queries from changes of users' requests.

6 Implementation of a Prototype System

Prototype system **GeoProxy** was implemented to prove that the dynamic map systems are feasible and useful compared with conventional layer-based map systems. **GeoProxy** are implemented by taking an object-oriented approach. The ObjectWorks\Smalltalk version 4.1 and the VisualWorks are used as the programming environment.

6.1 Object Specification and Importance Level Control

There are three major factors in order to change the contents of dynamic maps.

1. the latest data (Unlike conventional geographic systems, we assume that the most up-to-date data are always used to generate maps.)
2. query (Purposes of map generation are specified by queries.)
3. visualization method (Even if sets of objects to queries are determined, the resulting map cannot be uniquely determined, since the size of a map has the limitation in the number of geographic objects to be displayed.

GeoProxy allows users to control importance levels of classes, which are results of ad hoc queries (Figure 8). By controlling importance levels, instances included in more

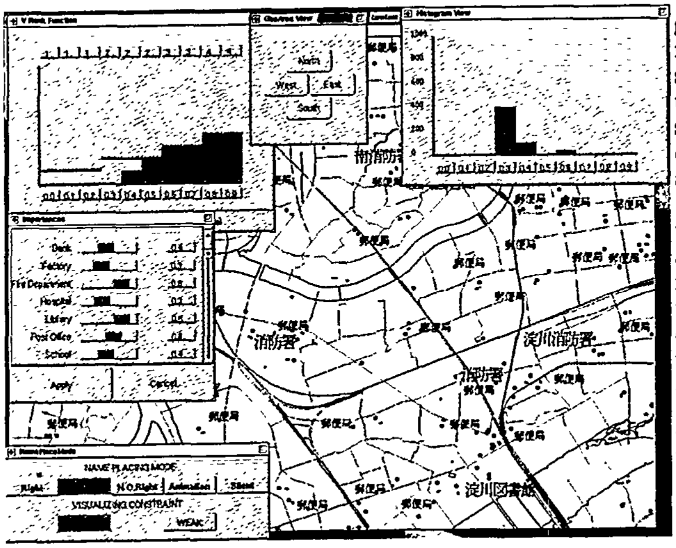

Figure 8: Importance Level Control

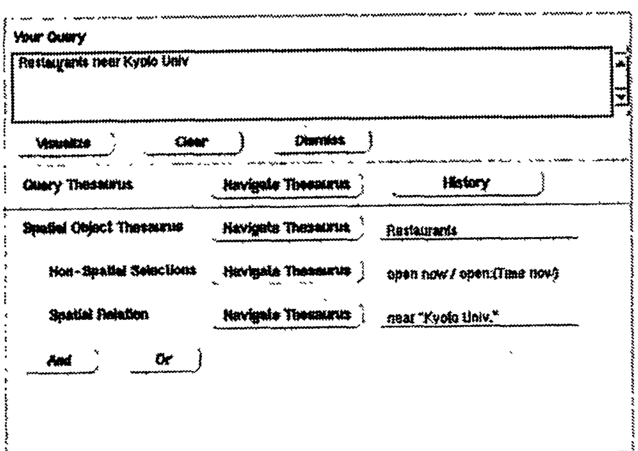

Figure 9: Editing an Initial Query

higher classes can be visualized as more informative graphical objects, e.g., large font, bright characters and so on.

6.2 Structured Editor

By the structured editor, words are derived from the geographic domain thesauruses and the query database, and an initial query is generated (Figure 9).

7 Conclusion

In this paper, the methodology which supports users to formulate geographical queries has been discussed. In the case of hypermedia based on database views, most of its components are derived from objects stored in databases. In addition to the objects, our framework of hypermedia is based on database with conceptual thesauruses, example queries and inferential mechanisms in order to visualize geographic information, i.e., maps.

Our system unifies geographic database systems and supplementary databases, e.g., thesauruses, as extended geographic database systems. Although this approach cannot be acceptable to all architecture of geographic database systems, proposed map systems are much more flexible than conventional layer-based map systems. Since our system manages fact and knowledge data separately, geographic data and their candidates can be easily searched and displayed by utilizing quite simple user interface.

Knowledge about cartography should be used for synthesis of feasible maps. Although deductive approaches are essential for geographic database systems, there are many problem in geographic information systems. Since the systems can not answer "Yes" or "No" to many queries, constraint-based approach is also essential. We considered that geographic database systems can extended by these approaches.

References

[1] M. McDonald, "Developments in Traffic Management, Information and Route Guidance Systems in Europe", Vehicle Navigation & Information Systems Conference Proceedings, IEEE CATALOG #94CH35703, pp. plenary-11 – 19, Aug. 1994.

[2] "The 1994 Plan for the National Spatial Data Infrastructure - Building the Foundation of an Information Based Society", Federal Geographic Data Committee, Mar. 1994 (ftp://fgdc.er.usgs.gov/gdc/general/documents/nsdi.plain.1994.ps).

[3] B. Plewe, "The GeoWeb Project - Using WAIS an the World Wide Web to Aid Location of Distributed Data Sets", WWW-Fall 1994 Conference, 1994 (http://wings.buffalo.edu/geoweb/).

[4] K. Hughes, "Entering the World-Wide Web: A Guide to Cyberspace", Honolulu Community College, Oct. 1993 (http://www/hcc/hawaii.edu/guide/www.guide.html).

[5] S. R. Newcomb, N. A. Kipp, V. T. Newcomb, "The "HyTime" Hypermedia/Time-based Document Structuring Language", CACM, 1991, Vol. 34, No. 11, pp. 67 – 83.

[6] N. Roussoupoulos, C. Faloutsos, T. Sellis, "An Efficient Pictorial Database System for PSQL", IEEE Transactions on Software Engineering, Vol. 14, No. 5, pp. 639 – 650, 1988.

[7] H. Aonuma, H. Imai, Y. Kambayashi, "A Visual System of Placing Characters Appropriately in Multimedia Map Databases", Proc. IFIP TC-2 Working Conf. on Visual Database System, 1989, pp.525-546.

[8] Masatoshi ARIKAWA and Yahiko KAMBAYASHI, "Dynamic Name Placement Functions for Interactive Map Systems", The Australian Computer Journal, Vol. 23, No. 4, November 1991, pp. 133 - 147.

[9] M. Arikawa, H. Kawakita, Y. Kambayashi, "Dynamic Maps as Composite Views of Varied Geographic Database Servers", Proc. of 1st Int'l Conf. on Applications of Databases (Witold Litwin, Tore Rische Eds.), Lecture Note in Computer Science 819, Springer-Verlag, pp. 142 – 157, Jun. 1994.

[10] A. H. Robinson, R. Sale, J. Morrison, "Elements of Cartography", John Wiley & Sons, 1978.

[11] J. Aitchison, A. Gilchrist, "Thesaurus Construction: A Practical Manual, 2nd Edition", Aslib, 1987.