Research in Informatics

Engineering Information in Data Bases and Knowledge Based Systems

Techno-Data '90

edited by
D. Richter, H. Grabowski

Volume 3

AKADEMIE-VERLAG BERLIN

Herausgeber:

Prof. Dr. Dieter Richter
Institut für Informatik und Rechentechnik
Berlin

Prof. Dr. Hans Grabowski
Universität Karlsruhe

Die Titel dieser Schriftenreihe werden vom Originalmanuskript der Autoren reproduziert.

ISBN 3-05-501272-0
ISSN 0863-4300

Erschienen im Akademie-Verlag Berlin, Leipziger Str. 3-4, Berlin, 0-1086

Printed in Germany
Gesamtherstellung: GAM-Media GmbH, Berlin
Lektor: Dipl.-Math. Gesine Reiher
LSV 1095
Bestellnummer: 764 215 8 (2192/3)

GAL: AN ECONOMIC GEOGRAPHY MAPS DATABASE SYSTEM. FUNCTIONAL DESCRIPTION

Peter Stanchev and Valery Vutov

Institute of Mathematics
Bulgarian Academy of Sciences
G. Bonchev Str., 8 - 1113 Sofia, Bulgaria
Telex: 22628 BANMAT BG, Fax: 359 - 275 9012

The paper is devoted to the description of a computer aided system in economic geography. The system includes possibilities for automatic derivation of economic information on the basis of scanned economic geography maps, stored structured information, and easy to learn data retrieval language.

The automatic way of information extraction and its storage in a structured form makes the system valuable for information retrieval and/or learning purposes.

The system differs from the existed scientific and commercial specimens by the mechanisms of information storage and retrieval, which permit the volume of the stored information to be significantly decreased (only a region-defining map and a set of relation tables are kept), and when required, the particular map image is recreated.

The system is presented by a description of its global architecture and main functions.

1. INTRODUCTION

When large amounts of economic geography maps have to be managed in a computer system, the need to apply the database technology naturally arises.

The query language and the access structures implemented in the database management systems are very powerful for retrieval operations. But in our case their potentiality is limited by the fact that the content of the maps has to be described using models which are developed for database systems rather than for maps systems, and so they lack the expressive power needed for maps manipulation. In fact, maps are inherently different from database records, their records can be divided in different classes according to their interpretation. The record structure in database systems can be described at class (i.e. type) level. Since the ratio of instances per type is very high, the resulting storage structures and access methods are very efficient. On the contrary, each map may have its own particular structure, and a whole semantic network [1] may be necessary to completely describe each map instance.

In the field of map data processing, several work have been reported. Some of them concern technologies for building geographical database systems. A geographical road database system project has been described by Olsson [2]. Siekierska [3] has presented an experimental graphic work station with different cartographic functions, which allows the user to manipulate existing maps and to create new ones. Kasturi and Alemany [4] reported a project for a system that automatically extracts information from paper-based maps.

The problem of map recognition is also treated in the literature. An automated map recognition system has been proposed by Ejiri et al. [5].

The type of the map database query language is investigated using different approaches. A QBE-like language is described by Barrera and Buchmann [6] for a geographical database system. Roussopoulos and Faloutsos [7] proposed a SQL-like language for managing pictorial database system.

In this paper we address the problem of building an economic maps database system (GAL).

GAL is a tool for introducing geography maps and geographical information into the computer, analyzing maps, in order to obtain their semantic content, storing the extracted information and retrieving different kinds of geographical data.

In GAL, the maps are accessed by providing: (1) a scanned region-defining map, which is the pattern map of the observed region; (2) a collection of economic information relation tables, automatically obtained during the analysis of a set of scanned economic maps, representing the observed region; (3) the scanned map legends.

2. SYSTEM FUNCTIONS

The main functions performed by the system are the following:

- data collection, physically performed in two ways, by a scanner and by keyboard;
- analysis of entered data, which are scanned maps with economic signs; it is performed for the purpose of obtaining structured data, reflecting the information represented by the map;
- storage of the region-defining map in the form of bit image array, and the structured information, in the form of a relational table;
- query management;
- output management.

The global system architecture is shown in Fig. 1.

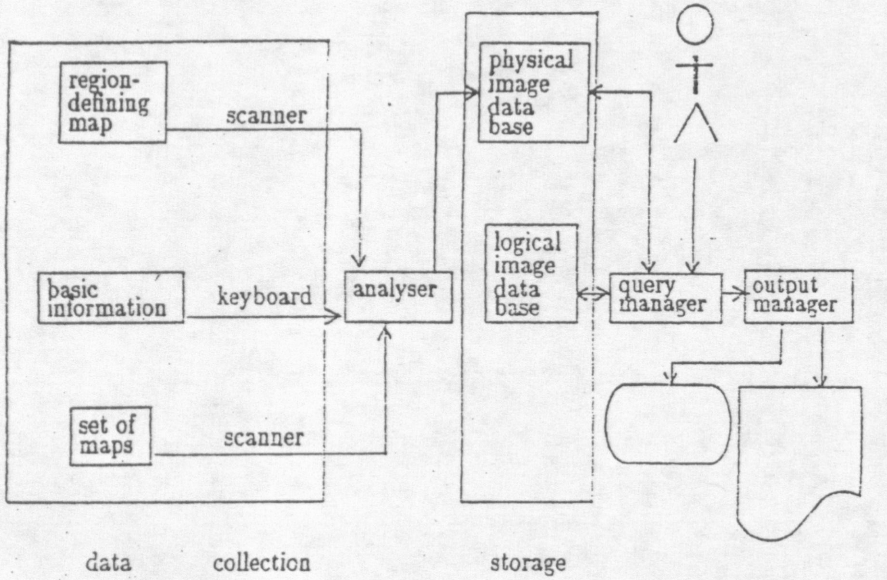

Fig. 1. *The system architecture*

2.1. Data collection

To make GAL functional we need the following data:

- region–defining map. This is the pattern of the observed region. We enter the region–defining map using a scanner;
- the scale factor for the legend signs. In the domain of economic geography three sizes of the legend size are usually used. They express the different degree of quantity;
- basic economic information. A collection of economic maps, for one and the same region, usually has some information that is common for all maps. The information is in the form of tables, which contain data such as: town names, population. area, geographical coordinates, road length, etc.;
- scanned economic maps and the corresponding legends.

2.2. Analysis of entered data

During the map analysis, an map is accepted and its elements (signs from the corresponding legend) are recognized, as well as their relative positions and their size.

The map analysis is based on the concept of graph matching using a particular indexing mechanism, which is described in details in [8], and is divided in two phases.

The first one includes the legend sign analysis. The scanned legend sign is converted to a graph [9], which is a two dimensional line representation of the scene. The process of conversion includes: (a) edge detection, (b) line segment encoding, (c) line approximation and (d) relation recognition. The result is an Image Table, composed of the set of recognized primitives (with their positions into the image) and the set of relations between them. Using the Image Table and excluding the information for the specific positions of the primitives, we obtain the Legend Sign Table. Two accessory data structures, Table and Global Indexes, are constructed. The Table Index consists of recognized relations and the Legend Sign Table identifications where these relations are located. The Global Index includes the legend sign features (primitives and relations between them) that uniquely identify the sign.

The second one includes the legend sign recognition in the particular economic map. The Image Table construction mechanism for the map is the same as in the case of legend sign image. The recognized primitives and relations between them are looked for in the Global Index. If we find them, we use the specific Legend Sign Table and the Image Table to eliminate the sign elements from the Image Table. If we do not find them, we search the recognized relations in the Table Index and try to find the corresponding table among the Legend Sign Tables that contain the specific relation value.

We find translation offset parameters for each element found and convert them into geographical coordinates.

2.3. Storage of the region–defining map and the structured information

The system store the information in two ways.

The region–defining map is stored as a bit image array.

The basic information, and the information resulting from the analysis of the economic maps are stored in an uniform relational table that contains the following information: the image identifier; the object name; the number of the object in the image; two arrays, with attribute names and attribute values; geographical coordinates of the object center, or list of boundary coordinates. Some objects have standard attributes. For example the object "road" has attributes "FROM", "TO" and "LENGTH". We represent the region boundary by a list of "(X,Y)" coordinates. Thus, our database consists of a bit image array and tables that represent the useful information extracted from the economic maps. These tables are filled automatically by the analysis of the maps and we have the possibility to update them manually.

2.4. Query management

According to the proposed query language the user specifies the image condition in the form:
<object_attribute> 'WHERE' <object_clause> .

The <object_attribute> is the name of an attribute, whose value we want to known.

The <object_clause>, which is used to determine whether a record is to be selected for processing, is a boolean combination of <object_group>s. The <object_group> is an expression of <object_attribute>, <relationship> and <value>, where:

- the <object_attribute> must be among the existing in the table;
- the <relationship> must be appropriate to the type of the field;
- the <value> must be appropriate to the type of the field. If the <object_attribute> is a numeric field, the <value> may be a number, e.g. 45, -34.4. If the <object_attribute> is a string field, the <value> may be a literal string, i.e. a sequence of characters enclosed in quotes e.g. 'Sofia'. For <object_attribute>s of either type, the <value> may be the name of another <object_attribute> of an appropriate type.

There must be at least one space between each of these three elements.

An <object_group> has a value TRUE, if the content of the stated <object_attribute> bears the stated relationship to the testvalue. An example of <object_group> is:

$$name = 'Sofia'.$$

The value of this expression is TRUE, if the value of the <object_attribute> "name" is identical to the string of characters $'Sofia'$.

Another example is:

$$area > 200.$$

The value of this expression is TRUE if the value of the "area" <object_atribute> is greater then 200.

The following relationships are available for testing the numeric <object_attributes>: "=" equal to; "<>" not equal to; ">" greater than; "<" less than; ">=" greater than or equal to; "<=" less than or equal to.

The following relationships are available for testing the string <object_attributes>: "=" equal to; "<>" not equal to; "<" precedes - TRUE if the value of the <object_attribute>; precedes the value in alphabetical order; ">" succeeds - TRUE if the value of the <object_attribute> succeeds the value in alphabetical order; "!" substring - TRUE if the value is a substring of the value of the <object_attribute>.

If the characters "?" and "*" occur in he string literal value, they have the following effect: "?" will match any single non-null character; "*" will match any number of arbitrary characters including possibly none.

Several <object_group>s may be combined by means of ANDs and/or ORs to form a compound condition. AND is an operator which binds more powerfully then OR. Brackets may be used to override this, if required.

The query can contain also a function name. The function DISTANCE measures the distance between two objects. The function DISTANCE_ROAD – the road distance between two objects. AREA measures the area of an object. The COORDINATE function gives the geographical coordinates of an object. HOW_MANY gives the number of the requested objects on the map. MIN and MAX give an arranged list of a given number of objects.

DISPLAY function shows the result as a geography map.

For the beginners, the query may be composed, using a menu–driven form, that leads the user to the correct query, showing him the result.

All the stored images fulfilling the query constitute the query answer set.

The image query algorithm includes searches for images which fulfil the query in the table.

The formal description of the proposed language is given in Appendix 1.

2.5. Output management

The output of GAL (i.e., the answer of the user query) is performed in two ways. The user can see the answer in form of a text (simple table) or in form of a geography map. For a chosen map corresponding operations are available to display it on a screen or to print it in different colors, shapes, positions, etc.

We envisage the possibility to organize the image exchange via a local area network. The same output can be directed to another node of a local area network.

3. AN EXAMPLE

As an illustration of GAL's functions, we go through its phases with an example.

Suppose the system image database contains economic maps of Bulgaria. Suppose a region–defining map and some basic information, such as town names, geographical coordinates, important roads etc. have been entered. Suppose further, that the map, presented in Fig. 3. is among the maps entered in the system by a scanner device. The relational table (Fig. 2.) might be obtained as a result of the map analysis.

IMAGE_ ID	OBJ_ NAME	OBJ_ NUMBER	ATT NAME	ATT VALUE	X CORR.	Y CORR.
2	Fe	1	quantity	small	26.1	43.8
2	Fe	2	quantity	big	26.7	43.5
2	Cr	3	quantity	small	27.1	43.8
2	Mn	4	quantity	small	27.3	43.7
2	Terrigenous-carbonate complex	5	quantity	small	list	list
2	carbonate complex	6	quantity	small	list	list

Fig. 2. *Relational table with the information structured*

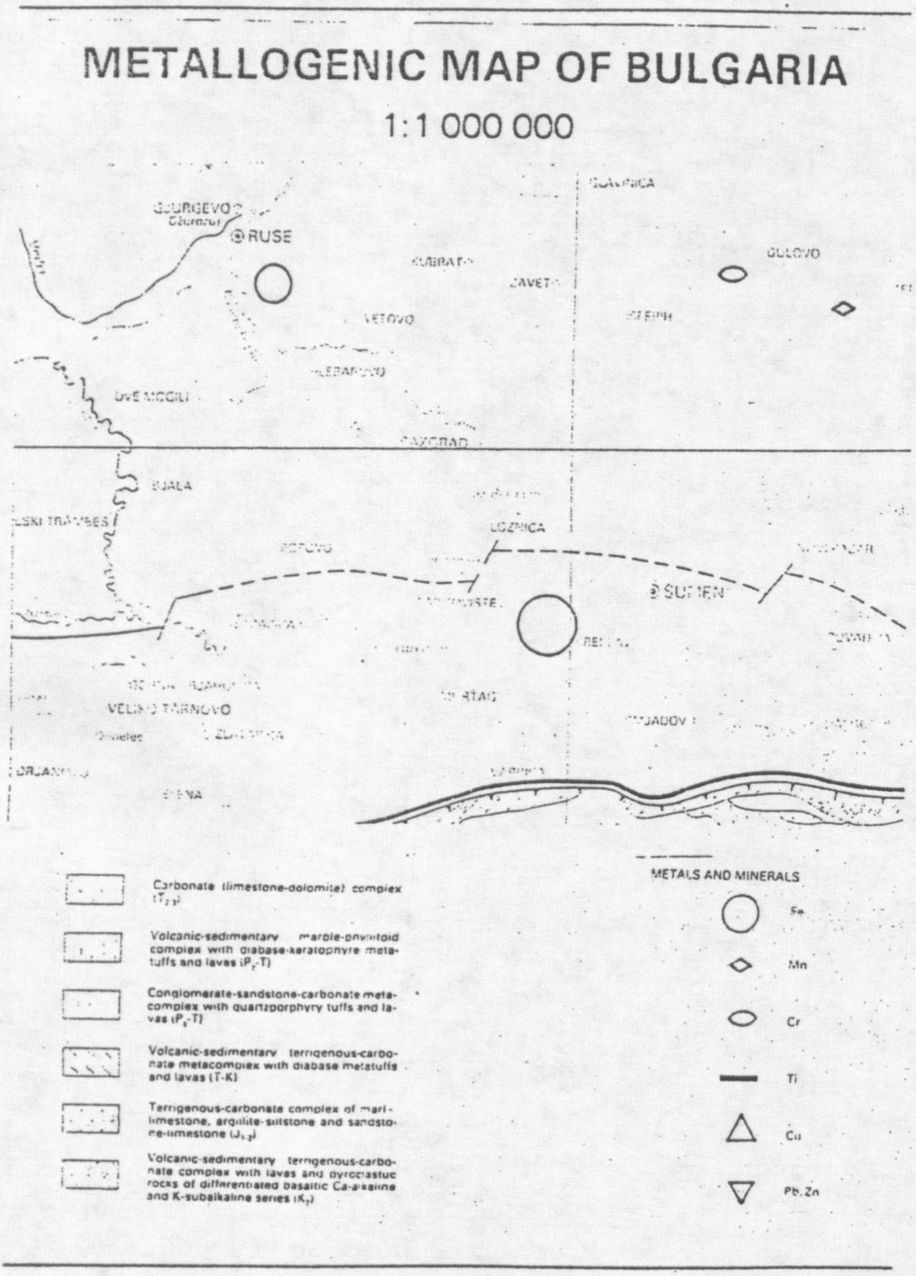

Fig. 3. *Metallogenic map of Bulgaria*

Some query examples concerning this map are the following:

Query 1. If the user wants to know the amount of Fe (iron) in the region to the east of the town of RUSE, the corresponding user query is in the form:

Fe.quantity WHERE Fe.XCORR > town.XCOOR AND town.name = 'RUSE'.

The result is:

small

big.

Query 2. If the user wants to know the area of the carbonate_ complex, the query is the following:

AREA(carbonate_complex).

The result is:

$243 km^2$.

Query 3. If the user wants to know the distance between the town of RUSE and the town of RAZGRAD, the corresponding query has the form:

DISTANCE(town.name='RUSE',town.name='RAZGRAD').

The result is:

64 km.

Query 4. If the user wants to know how many deposits of Mn (Manganese) there are on the map, the query is the following:

HOW_MANY(Mn).

The result is:

1.

4. ADVANTAGES

The main advantages of the proposed system are:

- the input of economic data is performed automatically. The maps and their corresponding legends are scanned, the analysis is performed, and the resulting data are structured into a relational table;
- the decreased volume of the occupied storage. Only the region–defining map and the legend signs are stored in a bit image format. All the other information is stored as a structured data;
- automatic reconstruction of the economic maps. Combining the region–defining map, the structured data from the relational table and the corresponding legend signs, the system automatically reconstructs the required economic map;
- simplicity of the logical data representation. For the user the table representation of the information is very comprehensive and easy to understand. That gives the possibility to learn, to compose correct queries in a very short time;
- simplicity of the query language. The query language is available in two forms – menu driven and direct forms. The first one is very helpful for the beginners.

5. CONCLUSION

GAL is under development on IBM PC computer. It is written in C and Assembler. It makes use of the Canon IX12 scanner.

A previous system prototype was implemented on IBM PC/AT computer and was intended for the management of business graphical images, generated by a commercial business graphics editor (IBM Graph Assistant). Another prototype described in [10], is applied for storage and retrieval of house furnishing design drafts.

6. REFERENCES

1. Tsichritzis D., Lochovsky F.,*"Data Models"*, Prentice-Hall, N.J. (1982).
2. Olsson L. *"Development of Road Database from Map Overlays"*, Nat. Defense Res. Ins. (FOA), (Dec. 1984).
3. Siekierska E., *"Toward an electronic atlas"*, Proc. Sixth Int. Symp. Automated Cartography, Ottawa, Canada, pp. 464-474 (1983).
4. Kasturi R., Alemany J., *"Information Extraction from Images of Paper-Based Maps"*, IEEE Trans. on Software Engineering, Vol. 14, No. 5, pp. 671-675 (May 1988).
5. Ejiri S., Kakumoto T., Shimada S., Matsushima H., *"Automatic Recognition of Design Drawings and Maps"*, Proc. Seventh Int. Conf. Pattern Recognition, pp. 1296-1305 (1984).
6. Barrera R., Buchmann A., *"Schema Definition and Query Language for a Geographical Database System"*, Proc. IEEE Conf. Computer Architecture for Pattern Analysis and Image Database Management, pp. 250-256 (1981).
7. Roussopoulos N., Faloutsos C., *"An Efficient Pictorial Database System for PSQL"*, IEEE Trans. on Software Engineering, Vol. 14, No. 5, pp. 639-650 (May 1988).
8. Stanchev P., Vutov V., *"A Model-Based Technique with a New Indexing Mechanism for Industrial Object Recognition"*, Ninth IEEE/IEMT Symposium, Washington D.C., USA, (Oct. 1-3 1990).
9. J. Buurma, G. Weller, F. Groen, G. Meyer and L. Hertzberger, *"Graph-Construction and Matching for 3-D Object Recognition and Pose Estimation"*, in "Artificial Intelligence and Information-Control Systems of Robots-89", I. Plander (edt.), North-Holland, pp.383-389 (1989).
10. Rabitti, F., Stanchev P., *"GRIM_DBMS: a Graphical Image DataBase Management System"*, IFIP TC-2 Working Conf. on Visual Database Systems, Tokyo, Japan, T. Kunii (edt.), North-Holland, pp. 415-430 (Apr. 3-7 1989).

APPENDIX 1.

```
query               : object_attribute 'WHERE' object_clause
                    : object_attribute 'WHERE' object_clause 'IN ORDER' 'A'
                    : object_attribute 'WHERE' object_clause 'IN ORDER' 'D'
```



```
                    : function_name
                    : DISPLAY '(' query ')'
                    ;

object_clause       : object_group
                    : '(' object_clause ')'
                    : object_clause 'OR' object_clause
                    : object_clause 'AND' object_clause
                    ;

object_group        : object_attribute relationship value
                    ;

relationship        : '='
                    : '<>'
                    : '>'
                    : '<'
                    : '>='
                    : '<='
                    : '!'
                    ;

function_name       : 'DISTANCE' '(' object_group ',' object_group ')'
                    : 'DISTANCE_ROAD' '(' object_group ',' object_group')'
                    : 'AREA' '(' object_name ',' object_group ')'
                    : 'AREA' '(' object_name ')'
                    : 'COORDINATES' '(' object_group ')'
                    : 'HOW-MANY' '(' object_name ',' object_group ')'
                    : 'HOW-MANY' '(' object_name ')'
                    : 'MIN' '(' query ')'
                    : 'MIN' '(' query ')' 'NUMBER' value
                    : 'MAX' '(' query ')'
                    : 'MAX' '(' query ')' 'NUMBER' value
                    ;

object_name         : STRING
                    ;

object_attribute    : object_name '.' attribute_name
                    ;

attribute_name      : STRING
                    ;

value               : INTEGER
                    : REAL
                    : STRING
                    ;
```

(65.) * Stanchev P., Vutov V., "GAL: An Economic Geography Maps Database System. Functional Description", ***Research in Informatics***, Vol. 3, Akademie-Verlag, Berlin, 1990, 197-206.