



## DEVELOPING MAP DATABASES: PROBLEMS AND SOLUTIONS

István Elek, Gábor Gercsák

Department of Cartography and Geoinformatics, Eötvös Loránd University  
(ELTE), Budapest (Hungary)

**Abstract:** This article is an introduction to a several-year long development focusing on the building of a digital map database, named EDIT [1]. The theoretical background, the applied technology, the developed information system, and its perspectives will be described. The classical relation database technology and the object oriented, document based database management [2, 3, 4] are applied beside those web based tools and technological solutions which are in the world of the WEB2. The system is not only a rich data source for researchers but also a virtual laboratory of the database technology where various database management systems related to GIS can also be investigated. The EDIT system has become an ordinary data source of the research and education at Eötvös Loránd University and its partners in Hungary [5]. The role of the system is of remarkable importance in the doctoral training too [6].

### 1. Introduction

Research projects and teaching at the university missed the availability of digital maps for long, although there were many paper maps in the map collection. A large number of map databases are now accessible in the Internet: some of them are free, but they mostly charge for their use [7, 8, 9, 10, 11, 12, 23, 24]. The laws regulating the use of maps are very different from country to country, and the data policy changes from government to government. It is not surprising that “sharp” maps are rarely found on free web pages. As for the fresh satellite images and aerial photos, the case is the same. It is even more difficult to find free vector maps, particularly those that have link codes to databases.

University researchers and teachers very much need a well-organized cartographic database that is available for everyone at the university. The Department has done important steps in this direction by setting up an information system named EDIT. (The acronym stands for the Hungarian words of University Digital Map Collection.) More than ten thousand maps have been entered into the database, which is completed with a data loading application and an online query system (Figure 1). Although the original plan only counted with users from the university, and the data loading is not complete yet, and a part of the users of the system already comes from outside the university. In the beginning, the EDIT system was only accessible from the nodes in the “elte” domain, that is for the teaching and research units that take part in the training of students of earth sciences and computer science. Later, the range of users greatly expanded. The University of Debrecen and the University of Pécs were the first to join the system. Further members include the University of Szeged, two Academic institutions (the Geographical Research Institute and the Balaton Limnological Research Institute), and the Institute and Museum of Military History.

This paper introduces the EDIT system and its major parameters.

## 2. University Digital Map Collection (EDIT)

The EDIT is an information system built on the principle of relation databases and made for the management, inventory, and service of raster and vector map data. The raster data include scanned maps, orthophotos, and hyperspectral aerial images, while the vector data are mostly available in ESRI shape and Mapinfo tab formats.

The logic and the background structure of the system, and the applications of the database provide the users with an easy and fine-tuned query, and a quick presentation of the results (alphanumeric and graphical or map data). Further, the access to the system can be easily regulated.

The system consists of two major parts: the EDIT is for data loading and maintenance, while the EDITKE is for online query (Figure 1). The online query system is accessible at <http://mapw.elte.hu/edit> for those on the elte domain. The clients of the Virtual Private Network (VPN) can reach this URL through the certificate of the VPN. The data maintenance application runs on the map server of the system.



Figure 1. First page of the online query system

The system was made in Microsoft VisualStudio 2008. The loading application is a win32 application. The online query was made in ASP.NET. István Elek developed both application programs.

### 3. Structure of the database

The great differences between the formats of the raster and vector maps caused difficulties for the planning of the database. Although there are various raster formats, they all contain an  $m \times n$  matrix, which stores the intensity values of a picture according to a colour model in rows and columns. These data may come from scanned paper maps or from the signals of digital images. They all follow the simple structure of raster data models. Descriptive data such as scale, time of making, sheet number can be added to the maps (namely, to each raster file).

This logic cannot be used for vector maps, because these maps are organized into layers and feature classes. Several types of information, layers, and feature classes together can only express the complexity of classical paper-based maps. In addition, some of the usual descriptive data (e.g. scale, title, and sheet number) cannot be interpreted for these data. It is obvious that different data structures must be used for storing the raster and vector maps due to their basic differences. Further, the vector data are often available organized in relation databases even though they are exported into file groups (e.g. ESRI shape or Mapinfo tab files). Consequently, the raster maps need a database structure completely different from that of the vector maps.

#### 3.1. Storage of raster maps

In most cases, the raster maps can be stored in a table structure in which one record represents one map with all its descriptive data as shown in the table below.

map_id	title	sheet number	scale	...
3455	Ödenburg	P48	300 000	...
...	...	...	...	...

The present large capacity database managers allow the binary storing of raster files, graphical data and various kinds of data in BLOB type fields. BLOB is an abbreviation of binary long object, which term expresses that it contains the original information converted into a series of bytes. To display the content of these fields is not as simple as that of the usual alphanumerical fields. However, a great advantage is its secure storage, because the authentication system of the database manager automatically protects the BLOB type data too. This is very important for the information systems that have a high risk of security.

However, the map collections do not belong to the category of high risk. In addition, it may also happen that the raster files of the maps must be modified (e.g. due to noise filtering, improving the quality of the image, georeferencing),

when the storage in the BLOB fields would cause problems. Therefore, the maps are stored in another way. The descriptive data are stored in data tables, while the maps are stored in file systems outside the relation database. The descriptive tables contain the name of the maps only, which is a reference in the map\_id field. Therefore, if the raster file is modified in any way, it will be immediately visible in the system without changing the database.

The raster map data are stored in two data tables. One stores the groups of maps (table of groups, Figure 2), the other one stores the data records of maps (rastermaps) (Figure 3). Individual maps are sorted into groups by referring to the table of groups.

gr_id	nev
1	Atlaszok
2	Budapesti kataszteri térképek régről
3	Egyéb térképek
4	EOV 1:10.000-es topográfiai térképek georeferálva
5	EOV topográfiai térképek NEM georeferálva
6	Földgömbök
7	Gauss-Krüger térképek az 50-es évekből
8	Gauss-Krüger térképek a mából
9	Hadtörténeti Intézet térképei
10	Hazai szelvényezésű térképek
11	III. katonai felmérés térképei
12	I. katonai felmérés térképei
13	Kogutowicz-féle térképek
14	OSZK térképei
15	ELTE Térképtudományi Tanszéki térképtár térképei
16	Világtérkép sorozat
17	Ortofotók

Figure 2. Table of groups

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
cim	VARCHAR(255)			<input type="checkbox"/> BINARY	NULL
tema	VARCHAR(100)			<input type="checkbox"/> BINARY	NULL
meretarany	INTEGER			<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
vetuleti_rendszer	VARCHAR(100)			<input type="checkbox"/> BINARY	NULL
felbontas	INTEGER			<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
leltari_szam	VARCHAR(25)			<input type="checkbox"/> BINARY	NULL
ev	INTEGER			<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
kiado	VARCHAR(255)			<input type="checkbox"/> BINARY	NULL
tipus	VARCHAR(100)			<input type="checkbox"/> BINARY	NULL
tartalmi_leiras	VARCHAR(255)			<input type="checkbox"/> BINARY	NULL
letrehozoz	VARCHAR(255)			<input type="checkbox"/> BINARY	NULL
kozremukodo	VARCHAR(255)			<input type="checkbox"/> BINARY	NULL
forras	VARCHAR(255)			<input type="checkbox"/> BINARY	NULL
nyelv	VARCHAR(100)			<input type="checkbox"/> BINARY	NULL
jogok	VARCHAR(255)			<input type="checkbox"/> BINARY	NULL
kapcsolat	VARCHAR(255)			<input type="checkbox"/> BINARY	NULL
megjegyzes	VARCHAR(255)			<input type="checkbox"/> BINARY	NULL
grformat	VARCHAR(4)			<input type="checkbox"/> BINARY	NULL
csopoz	INTEGER			<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
map_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL

Figure 3. The structure of table rastermaps

This simple relation (of 1 : n type) link describes the grouping of maps. Several other field values may be chosen for this purpose from the collection (e.g. scale), but as the present project is a pilot plan, we decided to have a simple structure first.

### 3.2. Storage of vector maps

There are various kinds of vector maps, and they are built up on different logic. Their models are much more complicated than that the raster data models. This explains why the present 3.0 version of the EDIT system does not store the vector maps organized in relation tables, but in the file system in a hierarchical structure. When planning the system, its simple use was highly considered, and the table structure designed for the raster data was not used for the vector subsystem. Figure 4 presents a vector layer of the borders of blocks in Budapest, which can be accessed by a hierarchic control (Tree View) shown in the figure. The layers are sometimes grouped according to a region they belong to (e.g. Budapest), sometimes according to the name of data products (e.g. ADC-WorldMap).



**Figure 4.** Navigation in the vector subsystem with a hierarchic control

The layers of the vector subsystem are available in ESRI shape or Mapinfo tab formats. Although there are other file formats (e.g. ArcInfo personal database in mdb file formats), we decided that these two standard formats provide the system with enough flexibility, because every GIS software can now read or import these file formats.

### 3.3. Query in the database

You can search in the raster subsystem according to the SQL (structured query language) standard. The full functions of this kind of online search are only available for special users with the help of a series of SQL commands and to a very much limited extent for users without personal identification (such as queries from any elte domain). Naturally, the whole arsenal of SQL can be applied in the process of data loading and database building, because the data loading application supports these processes.

There is no SQL interface in the vector subsystem. The expressive names of libraries and files make the navigation easy in the file system. A table of metadata might be put into service with the pure task of querying in the future, but it is certainly not going to be a part of version 3.0.

## 4. Authentication, security and data loading

The authentication of the database against hackers in the cyber space is a very important issue. It is not worth building large databases if we cannot protect the server from the attacks of hackers, because the frequency and complexity of the attacks will surely lead to data loss.

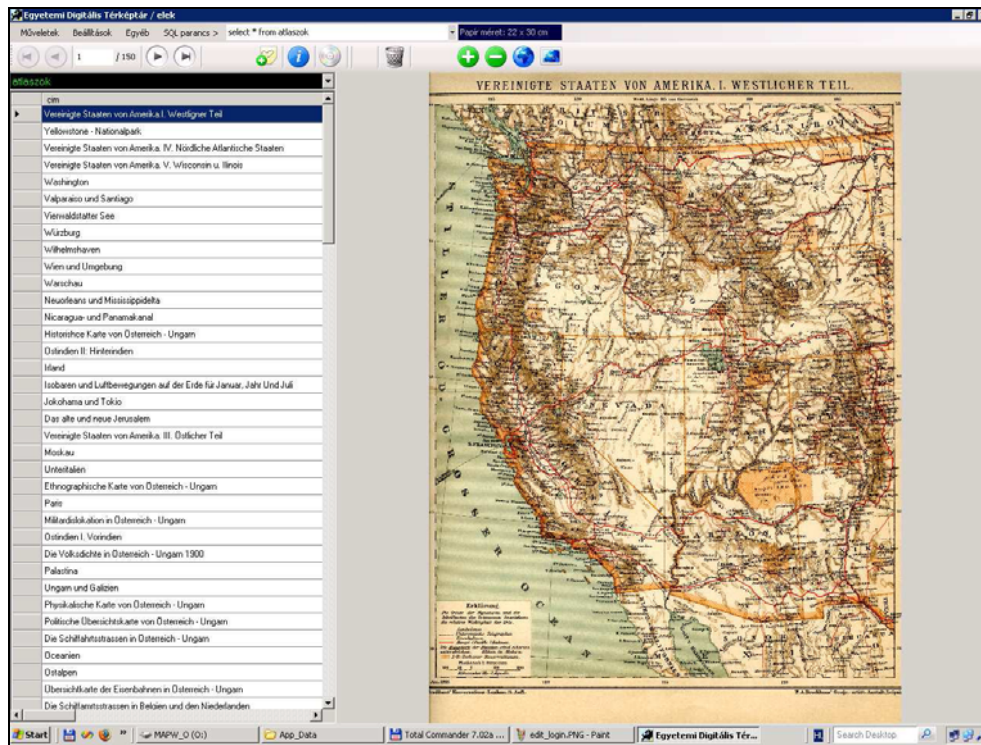
The protection must be guaranteed at two levels. Namely, the server and database must be protected. The protection of the server is the task of the administrator of the operating system, which includes the following tasks: virus protection, setting up of the firewall, setting up of the Internet Information Service, and the regular refreshing of the operating system of the server. The server is not accessible from a local network. It has only one public directory, into which certain files (e.g. tif, jpg, tab, shp) can be loaded from a few dedicated computers. In this case, the data loading application imports the requested maps.

Although the protection of the server protects the database too, but the efficient protection of the database needs additional settings. It must be decided in the administration of the database management system that which users have the right to access the database tables and to what extent. There are users who, without personal identification, have the select right only. (These are requests from URLs on the elte domain.) Others, who have personal authentication, have not only the select right to access the database. The series of SQL commands is also available for them, and they can download the selected maps on their client computer.

Only users with special rights are allowed to load and maintain the data. They have the right to not only select, but also to update and delete information. However, they do not have the right to delete large amount of data, to drop tables or to perform any other operation that would corrupt the data. The administrator only has full rights over the database.

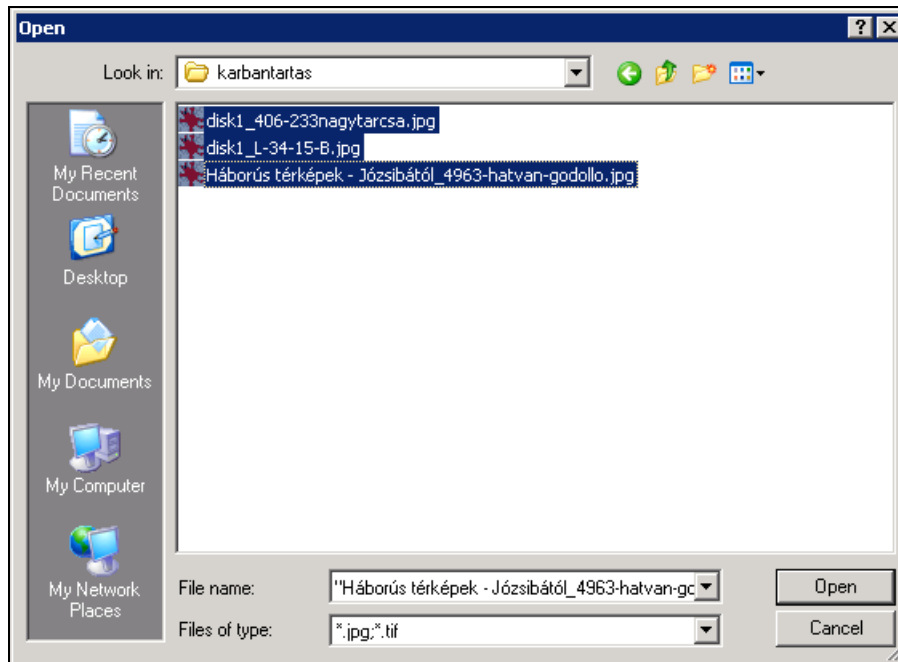
The loading and maintenance of data is performed by a win32 application running on a server, which is available for registered users only. This application allows the editing of the existing map data, the changing of the attribute data, as well as updating the data and the manipulation of the existing maps (Figure 5). In addition, the new maps are entered into the system by using this program

(Figure 6). The directory of the server that receives the uploaded data is only accessible with the right of writing from a few client computers. The map data to be uploaded into the database are placed here, in the buffer. After loading the map into the database, the buffer becomes free. The configuration file of the system includes, among others, this place as well, which is specified at the setting of the system.



**Figure5.** After starting the data loading application, you enter the database. You can edit the data of a selected group of maps and you can move and select in the table to view the chosen map. Browser and mapped windows help view the data.

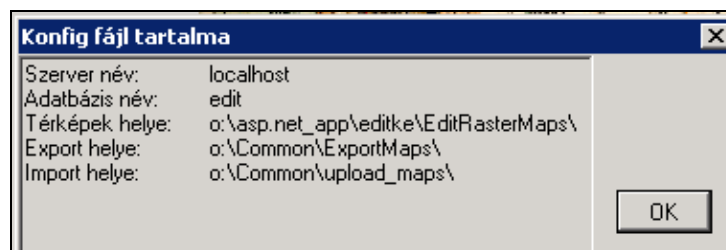




**Figure 6.** The data loading application can import not only single maps, but also groups of maps into the system

## 5. Query and servicing

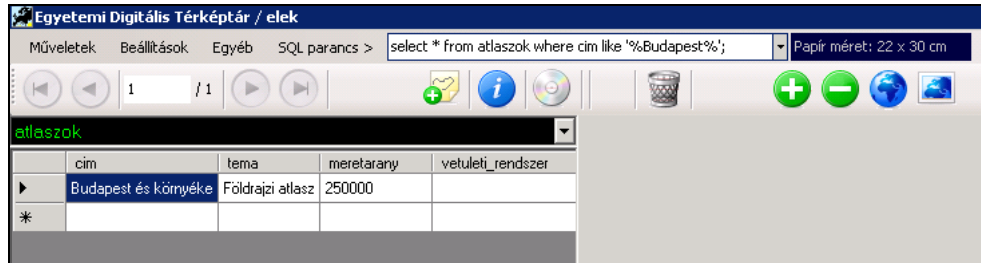
The system includes a configuration file, which contains the parameters necessary for the operation (Figure 7). This file ensures that the places that are needed for the access to and management of the data and maps are flexibly specified. This guarantees the smooth running of the system even if the operation system is completely reconfigured. In this case, the accessibility and addresses of the modified places have to be rewritten in the config file only. There is another parameter file, which gives the names of the data columns to be displayed online. This is necessary because not every user is interested in every data column. This parameter file allows us to specify which data columns should be visible for the online query.



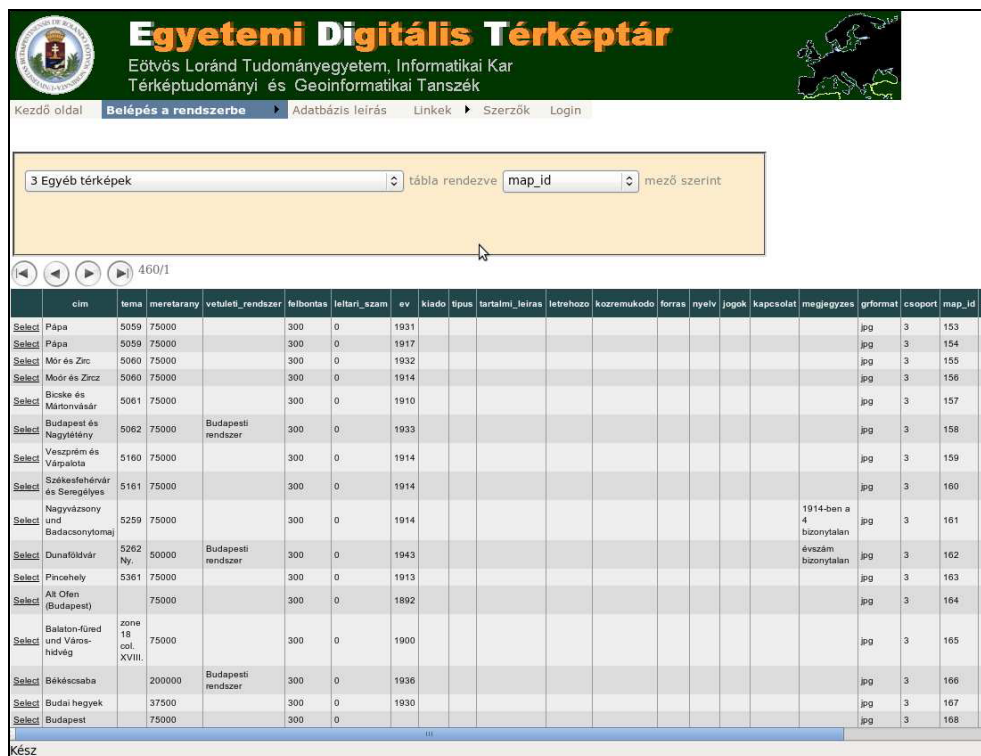
**Figure 7.** The contents of the configuration file of EDIT



When uploading data, complicated queries may be necessary. This is offered by a series of SQL commands, which allows the user to make a combination of SQL commands at wish (Figure 8).



**Figure 8.** In the series of SQL commands you can give optional commands such as find all those raster maps that have the word Budapest or 1 : 100 000 in the title

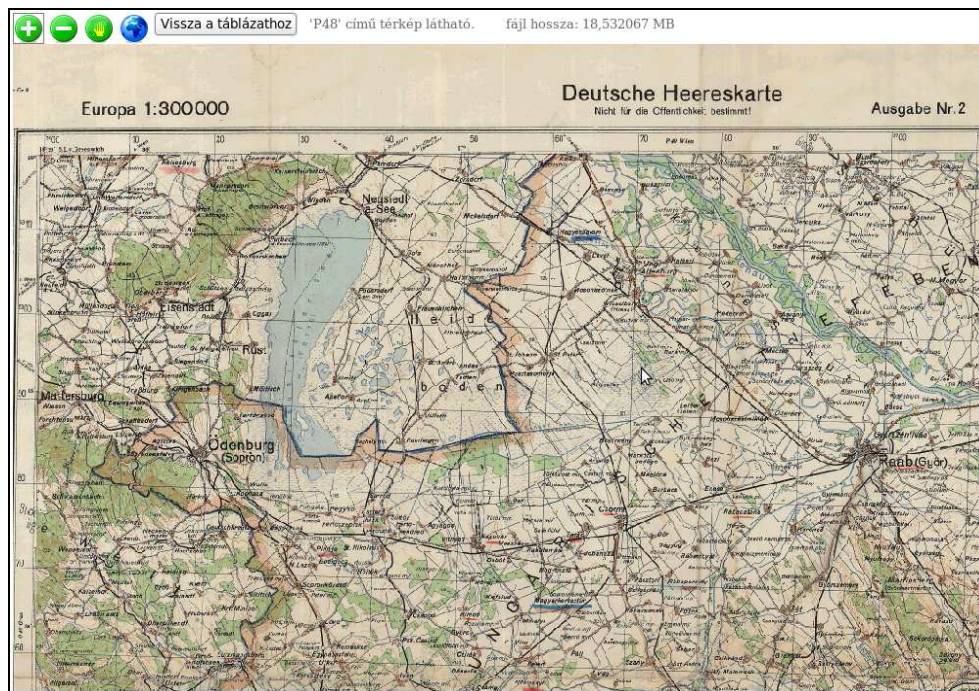


**Figure 9.** The screen image of EDITKE shows the attribute data of the first twenty maps in the egyéb (other) group of maps arranged according to the field map\_id. The SQL commands are not visible here, because the user did not have the necessary authentication.

The data uploading application offers a function that allows us to copy the selected map into a specific place. (Its role is very much similar to that of the Download function of the online query.) The user can copy the map into any place from here. One of the major aims of EDIT system is to provide the re-

searchers and students with free access to any map stored in the system. This place, among others, is included in the configuration file of the system.

The web and the database technology together led to the building of an on-line search and query system, EDITKE (KE stands for the Hungarian word *ke*res). The program in ASP.NET is an application of the server, which displays the data of the maps that satisfy the given conditions (Figure 9) and the selected map (Figure 10).



**Figure 10.** Viewing the selected map. The downloading function is active for the user with authentication, who can now download the map on a local client computer.

## 6. Conclusions

The latest EDIT system is version 3.0, and it has been more or less continuously developed. More and more research institutes and universities outside Budapest use it. Several interested students of informatics and cartography choose the problem of developing large map databases as their research topic.

This is an extremely rich topic, although the problems seem to be simple. Namely, we have the digital maps completed with descriptive data. Organizing them into a relation database is not a difficult task. The size of the raster maps can be very large (between 25 MBytes and 500 MBytes), while the vector maps have very varied structures. The size of a hyperspectral image can be extremely large (even 4–4.5 GBytes). As a result, the total size of the images for a certain area may exceed 1 TByte (1000 GByte). Handling such a large amount of data is not easy at all, not to mention displaying them. Displaying the images of several

megabyte size on the Internet is particularly problematic, and moving these images on the web is really difficult. It is necessary to develop special programs (services) on the server side that generate such images the size of which do not exceed the size offered by the resolution capacity of the given video card. This is the way to eliminate the barriers caused by the differences in the bandwidth.

The development must consider some other aspects that are not included in the present 3.0 version. One of these aspects is the transfer to or at least the study of transferring to object-oriented database management (e.g. the use of MongoDB) in the map server. Another aspect is the tracking of changes in time. The relation database managers always face the difficulty of tracking the changes in time, particularly if complicated graphic data (such as maps) change.

The present EDIT system runs in Windows server environment (Windows server 2008, Internet Information Service, VisualStudio, ASP.NET), but it would be interesting to test it in Linux environment (Linux, Apache, Postgresql, Java). The increasing popularity of open source systems cannot be stopped: therefore, limiting the system to Windows environment would not follow the trend.

These are important questions, because the EDIT map collection is an experimental system, the terrain of informatical and cartographic experimenting, where all kinds of new technologies, algorithms and ideas can be tested.

The project is supported by the European Union and co-financed by the European Social Fund (grant agreement no. TÁMOP 4.2.1/B-09/1/KMR-2010-0003).

## References

- [1] I. Elek: Nagyméretű térképi adatbázisok fejlesztési és működési tapasztalatai., GEOGRÁFIA–2010–PÉCS: V. Magyar Földrajzi Konferencia. Pécs, Magyarország, 2010.
- [2] Zs. Máriás, B. Dezső, R. Giachetta, I. Elek: Cartographic Symbol Detection Using Local Segmentation and Other Methods. In: Bhanu Prasad (Ed.) International Conference on Artificial Intelligence and Pattern Recognition (AIPR-10). Orlando, USA, 2010. 07. 12–2010. 07. 14. Orlando: Paper 196.
- [3] R. Giachetta, I. Elek: Developing an Advanced Document Based Map Server. 8th International Conference on Applied Informatics. Eger, Magyarország, 2010.
- [4] I. Elek, R. Giachetta, Zs. Máriás: Egyetemi digitális térképtár fejlesztése az ELTE-n. In: Kákonyi Gábor (szerk.) Fény-Tér-Kép konferencia. Dobogókő, Magyarország, 2009. 11. 12–2009. 11. 13.
- [5] I. Elek, B. Kovács, K. Verebiné Fehér: Digitális térképtár az ELTE-n. Magyar Földtudományi Szakemberek VII. Világtalálkozója, Magyarhoni Földtani Társulat, p. 80., 2004.

[6] P. Rigaux, M. Scholl, A. Voisard: Spatial Databases with Application to GIS. Morgan Kaufmann Publishers, 2002.

Relating websites:

[7] Map server of Berkeley University :  
<http://www.lib.berkeley.edu/EART/topo.html>

[8] Web pages with cartographic content at the homepage of Department of Cartography and Geoinformatics, Eötvös Loránd University:  
<http://lazarus.elte.hu/hun/index.html>

[9] Web pages with special cartographic content at the homepage of Department of Cartography and Geoinformatics, Eötvös Loránd University:  
<http://lazarus.elte.hu/gb/linkek.htm>

[10] Web page of NASA: [www.nasa.gov](http://www.nasa.gov)

[11] David Rumsey's web page <http://www.davidrumsey.com/>

[12] Web pages with special cartographic content at the homepage of Stanford University:  
<http://library.stanford.edu/depts/branner/collections/sovietmil.html>

[13] Web page of Terraserver: <http://www.terraserver.com/>

[14] Hiking routes: <http://turistautak.hu>