

# Developing an Advanced Document Based Map Server\*

Roberto Giachetta<sup>a</sup>, István Elek<sup>b</sup>

<sup>a</sup>Department of Software Technology and Methodology  
Eötvös Loránd University  
e-mail: groberto@inf.elte.hu

<sup>b</sup>Department of Cartography and Geoinformatics  
Eötvös Loránd University  
e-mail: elek@map.elte.hu

## Abstract

Digital maps are the most important data source of Geographical Information Systems, and publishing of maps is an essential governmental task. For several years, the ELTE Faculty of Informatics has been developing a digital map database server known as EDIT (short for University Digital Map Server in Hungarian) with the foremost aim of assisting university scientific research projects and education by providing online access to various digital raster and vector maps, aerial and satellite imagery.

This paper presents the ideas, principles and practical aspects behind the development process of this system. This system will offer novel features compared to present map servers, through an interactive gateway and a secure network service. For that purpose, the entire system structure has been redesigned from scratch, with new data handling concepts.

*Keywords:* digital mapping, online access, document-oriented databases, image manipulation

*MSC:* 68P20

## 1. Introduction

Digital maps are the fundamentals of modern day cartography, and with it, Geographical Information Systems (GIS). Digital maps can be created by either digitizing printed maps to create raster images, or by using some sort of vector based mapping software. One would think that the previous method is obsolete in the 21st century's digital environment, but there are still many printed maps, that have not yet been digitized, although they still hold value to them. Certainly the most

---

\*The research is supported by the KMOP-1.1.2-08/1-2008-0002 project.

valuable these maps would be, if not only a raster, but also a vector representation could be obtained, to enable further editing or the use of spatial operators.

Automated vectorizing of raster maps is not an easy task, as there can be many problems, which hinder this operation. However in recent years notable steps have been made in this direction at the Eötvös Loránd University, Faculty of Informatics, and a dedicated system, known as IRIS has been created for this purpose. Details of this system can be read in [1] and [2]. During this research, the stock of scanned raster maps has grown vastly and the need for a storage system was formulated, so development has begun on EDIT. The name came from Egyetemi DIGitális Térképtár, which is Hungarian for University Digital Map Server. This project had the goal of providing a fast storage and retrieval application that can hold all scanned raster maps on one server. Since then, the EDIT project has exceeded the single purpose of being the data source for research done under the IRIS project, and it has become the mainframe of the Department of Cartography and Geoinformatics for storing digital maps (also in vectorized form) and remote sensed images.

Because the maintenance and publishing of maps, aerial and satellite images is an important task in of the government sector, and can also be found in the private sector, online spatial data sources like EDIT are required in many fields outside the University. However, the solutions that exist usually have limited capabilities. In this paper we present the goals and ideas behind the next development phase of the EDIT project, which will be a major step for our system, and also notable in the field of map server applications, as it will encapsulate solutions, that not have been presented in this area before.

The rest of the paper is arranged as follows. In Section 2 we will introduce the current map server solutions. In Section 3 the concept of our new approach will be clarified, and the realization will be presented in Section 4. We will conclude in Section 5.

## 2. Online map servers

Today, the fastest way of publishing any kind of data is through the World Wide Web, so it also became the primary platform for digital maps. Possibilities range from online purchasing of maps and satellite images (like TerraServer) to global map browsing with different search possibilities (like Google Maps and Bing Maps). The OpenStreetMap project enables users to edit the map contents; therefore it is probably the most adaptive online map.

Closer to our subject are the type of map servers that offer any kind of spatial content with multiple file format and database support. Probably the most well known open source solution is the UMN MapServer (see [3]), which runs as a CGI application on a web server, and provides native interface for browsing, downloading and uploading of maps with support for projection system conversions. It is compatible with many databases systems and can store vector and raster data in many forms, and also offers Open Geospatial Consortium (OGC) standard network

services for data retrieval (WMS, WFS, WMC and others).

Commercial solutions include ESRI ArcSDE (see [4]), which is part of the ArcGIS Server system, and is sort of a middleware between the popular relational database management systems (RDBMS) and client mapping applications (foremost ArcGIS Desktop). Besides providing a communications channel between ESRI applications and data stores, it also supports OGC standards, and can perform spatial operations directly on stored data. It can contain 3D spatial information using a unique storage format, and support parallel editing of spatial data with transaction semantics.

In our solution we would like to combine the benefits of the previous samples, and present an architecture, that can be easily extended with new features (like supporting new types of spatial storage, new raster image operators, etc.). However this is not the development course EDIT has been previously on.

## 2.1. The EDIT system

The development of the EDIT map server has begun in 2004, and one year later the first version was released. In this first development phase descriptive data was merged into a single MDB file with multiple tables for different groups of maps, and the images files of the maps were stored directly on the server file system.

With its GUI interface it soon became inadequate, and has been replaced with the second version using an online interface and a traditional RDBMS, specifically MySQL. This system managed to contain about 300 GB-s of raster and vector data of over 12 000 maps. Usage of the system has grown, and it has become the primary source of maps for the Department's research and cartography studies. Thanks to the internal usage of the system, there was no need for functionality beyond the upload and download of the maps, and search through descriptive parameters. Version 2.0 of the system can be seen on figure 1.

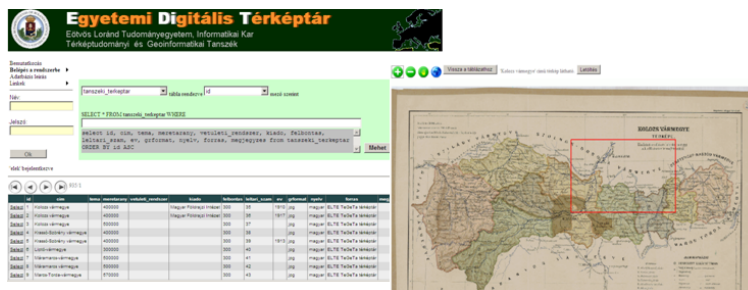


Figure 1: The online interface of EDIT 2.0

However, in the meantime several other groups and Departments have confirmed their interest in the usage of our map server. With possibility of opening the system to outside requests and sources many questions have risen, that not have been

previously treated. The systems needs to bear even more data (both graphical and descriptive) with different kinds of descriptive parameters, and needs to have much more detailed search options (not only through descriptive properties, but also through graphical, spatial and temporal properties). Using an RDBMS means using tables defined by the database schema, so when adding map descriptive data (used to query maps), the items in one category must all have the same descriptive data format. Due to the variety of maps, this limitation also needs to be overcome.

The most important task nevertheless, is the handling of copyrights, as most digitized maps, especially former government maps and remote sensed images come with still pending copyright protection, so it is important that no one, but the people legitimate according to the property rights are able to view or download the maps. These functions are not available in the current system, because there is no proper usage rights management.

### 3. Rethinking the concept

When preparing for the development of the next version of the map server, the limitations of the current system have been concerned, and a new goal has been laid down.

This goal is to develop a system, that can serve as a platform for any map related research work not only by providing data through online interface, but also providing a separate network services for other programs to directly browse, request, and modify data through accurate identification. These services should be implemented according to OGC standards (WMS, WCS, CSW, etc.). Since spatial operations and image processing can take much resource, modification operations need to be made on the client side. For the proper handling of copyrights, several usage right categories and identification methods need to be implemented.

The system should also be able to reach other online data providers, like Google Maps or NASA (employing proper usage right handling), so that data can be gathered from external sources. Detailed search needs to be available with methods to simultaneously display (using layers, see figure 2), compare and query spatial data based on location and temporal properties. For that, internal spatial data must be geographically and temporally located with the possibility of having any kind of descriptive data. The storage or any file format should be made uniformly by developing a single storage format for 2D and 3D data. Several import and export options must be provided for different forms of raster and vector data, and not only multispectral, but also hyperspectral images (which can contain up to several hundred spectral bands) should be supported. The application should be flexible enough so that it can be extended any time with new algorithms, import or export modules and data providers.

Under this brand new concept the entire structure of the system needs to be reconsidered, and redesigned from scratch.



Figure 2: For simultaneous display of maps layers will be introduced

## 4. The redesigned system structure

The structure of EDIT 2.0 is pretty simple. Only the database is used as a communication channel between the client side web interface and the administration interface. Both supply a data access layer for querying data with modification only available on the administrator side.

In the new concept there is a third interface beside the client and administrator environments providing network services through standard protocols. Separate core libraries will be introduced for holding the layers responsible for inner data access, user rights management, and data processing (see figure 3). An Application Programming Interface (API) will be available to extend the server with new functions and format support.

For the homogeneous and fast implementation of the system, Microsoft .NET technology will be used.

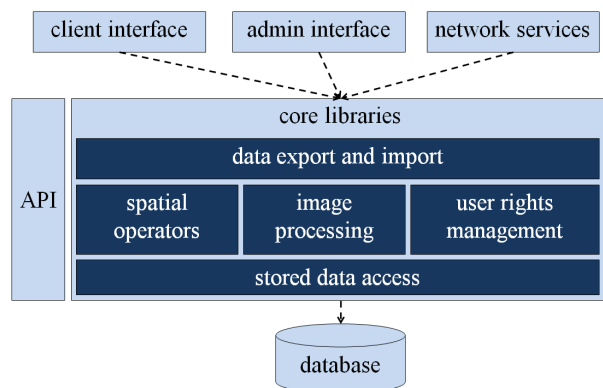


Figure 3: The proposed structure of the new system

## 4.1. Database background

The traditional relational database model has the lack of previously defining database and table schemas; therefore it is not appropriate for our goals of storing any kind of descriptive data with the spatial objects. Instead, the system will use a document-oriented database management system (DODBMS), which stores all items as documents. Documents are semi structured data objects, which can have different kinds of attributes, even other documents. XML datasets are standard example for DODBMS. However, in our approach we will use the MongoDB database server (see [5]), which stores the documents in collections, and does not need to define any previous schema information. MongoDB uses a separate internal file storage system, called GridFS, so files of any large size can be stored directly in the database.

In our system structure, collections will represent map categories, and special collections will be used for administration and logging (see figure 4). Input files will first be processed, raster images will be separated by spectral bands, and vector images will be disassembled by components. Since MongoDB does not support geospatial data storage, information retrieval from such dismantled map data will be implemented in the data access layer. The access layer will introduce spatial and topological operations usually offered by geospatial database extensions, like Oracle Spatial and PostGIS.

To support simultaneous editing and rollback functioning, version management of documents will also be implemented with locking and merging features. Users will be able to recall the history of any stored item, or invalidate any changes made. Proper usage right handling will also be implemented using different right levels and categories. Rights can be dynamically extended by introducing new categories.

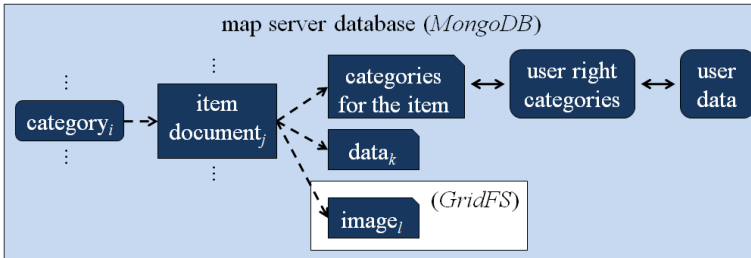


Figure 4: Database structure using MongoDB

## 4.2. Spatial and image operations

Due to this unified storage structure, implementing spatial and image operations can be done uniformly. All objects will be stored according to the WGS84 ellipsoid (which is the base of the GPS system), and there will be multiple projection systems

---

allowed. This is due to data loss in case of forcing only one projection system on all spatial data. When adding new objects to the server adequate descriptive data must be added, and reprojection or georeferencing may be done to adapt to the given project systems.

Besides georeferencing and projection systems, the operations layer will also provide spatial operators for matching, cutting, clustering, etc., and also provide standard image geometry and intensity transformations, like filters, pattern matching and segmenting of images.

The core libraries architecture will be established so, that the system can be dynamically extended with new algorithms for data manipulation using the API. For that several interfaces will be provided, so that using previously implemented image editing libraries (like GEO FilterBank, see [6]) will also be possible by creating a managed adapter for the server.

Note that the operations done by the clients must be done inside the client machine domain, using its hardware, so that the operations don't require any of the server machine's processing power. For that purpose, a client side display technology will be used.

### 4.3. Client and administration

The chosen display platform will be Microsoft Silverlight. Using .NET technology, Silverlight allows a programming environment compatible with the server core, and a much richer interaction through in-browser and out-of-browser GUI environments. It can use safely authenticated network channels, and automatically buffers images to accurate detail level. Using multiple canvases, the simultaneous display of maps can also be simply archived.

With the object movement safely preformed between client and server, all operations can be done inside the client environment, with the ability to save changes back to the server.

The admin environment will use a standard GUI application, enabling deep view of server status, document history, and logs. Naturally all client features will be available on the server side.

## 5. Summary and future goals

In the previous sections we have proposed the architecture of the next development phase of the EDIT map server. These innovations are not only new to the EDIT system, but also are fairly new to the family of map server applications. Using a document-oriented database system, client side technology, network services and dynamic extension capability, our application exceeds the capabilities of most map servers.

However, in this concept there are still many questions open. These include the automatic importing of items from the current database to the new structure.

Because most of the raster images are not georeferenced, and performing georeferencing manually is not manageable with over ten thousand items, automated georeferencing methods must be studied to find the optimal one suited for our scanned images.

## References

- [1] ELEK, I., Automatic generalization of maps by digital filtering, *GEOMATICS-RIGA*, Vol. 11 (2007), 64.
- [2] ELEK, I., DEZSO, B., MÁRIÁS, Zs., Image processing methods in raster-vector conversion of topographic maps, *International Conference on Artificial Intelligence and Pattern Recognition (AIPR-09)* (2009)
- [3] LIME, S., MapServer: Past, Present and Future, *Open Source GIS Conference, Second Annual MapServer User Meeting* (2004)
- [4] GASKILL, J., BROOKS, D. Understanding ArcSDE, *Annual ESRI International User Conference* (2000)
- [5] CHODOROW, C., Introduction to MongoDB, *Free and Open Source Software Developers' European Meeting (FOSDEM)* (2010)
- [6] SZENDREI, R., ELEK, I., FEKETE I., Texture based recognition of topographic map symbols, *International Conference on Artificial Intelligence and Pattern Recognition (AIPR-09)* (2009)