UMapIT: An On-Demand Web Mapping Tool Based On A Multiple Representation Database

Eveline Bernier, Yvan Bédard, Frédéric Hubert
Centre de recherche en géomatique, Université Laval
{eveline.bernier, yvan.bedard}@scg.ulaval.ca, frederic.hubert.1@ulaval.ca

1. Introduction

Web mapping applications are governed by the same rule as any other Internet application: response times must be very fast. In order to satisfy this requirement, we believe that web mapping applications must rely on Multiple Representation Databases (MRDB) as there is actually no complete solution for on-the-fly generalisation [Basaraner, 2004; Weibel *et al.*, 2002]. Besides the fact that they allow nearly instant answers, MRDBs also open the door for new navigation and customization functionalities for such applications. They are the centerpiece towards **on-demand** web mapping applications where the user produces a map that is more adapted to his specific needs and that support his needs for easy navigation and modification.

This paper presents UMapIT (Unrestricted Mapping Interactive Tool), an on-demand web mapping tool based on a specific MRDB. This database is structured according to the Vuel concept [Bédard and Bernier, 2002] and supports geometric, semantic and graphic multiplicities. The UMapIT architecture is first presented including a brief review of the Vuel concept. The functionalities that are today made possible by the MR management are then presented and we finish with future functionalities to be implemented.

2. UMapIT

2.1. Architecture

The actual version of UMapIT relies on an architecture that is based on interoperable standards such as XML, GML, SVG, WFS, from the OGC (Open Geospatial Consortium) and from the W3C (World Wide Web Consortium) (Figure 1). This architecture is composed of a data server (i.e. the application-specific Vuel database), Web services (i.e. the database querying services (descriptive and geometric features)) and a web client.

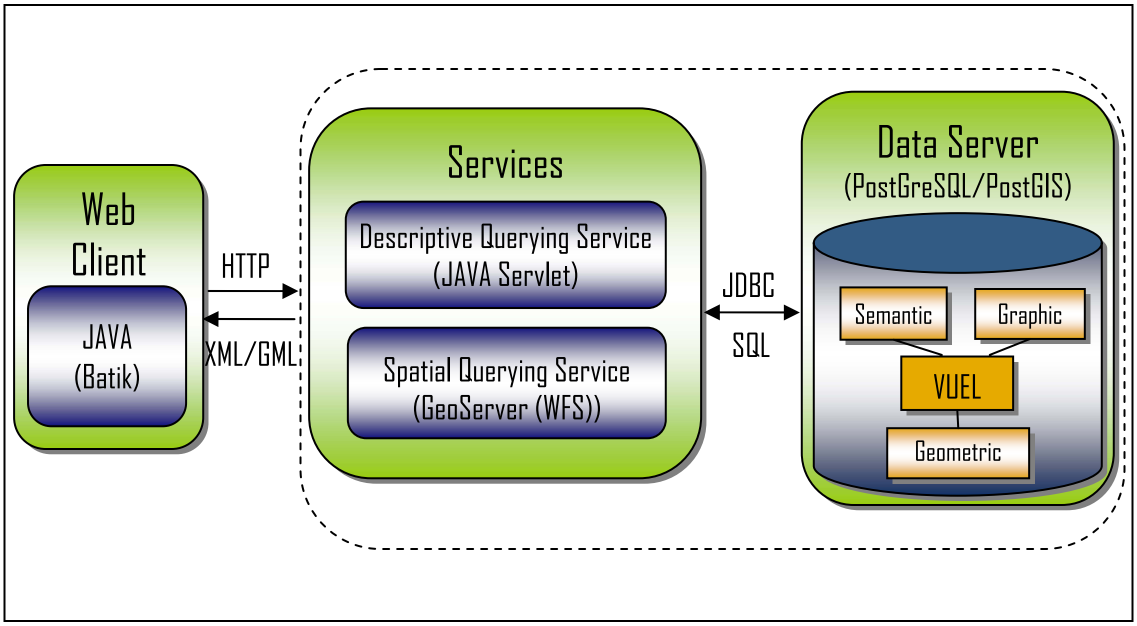
Web Client
JAVA (Batik)
HTTP
XML/GML
Services
Descriptive Querying Service (JAVA Servlet)
Spatial Querying Service (GeoServer (WFS))
JDBC
SQL
Data Server (PostGreSQL/PostGIS)
Semantic
Graphic
VUEL
Geometric

Figure 1 *The UMapIT architecture*

2.1.1. Data Server

The Data Server uses PostgreSQL/PostGIS to manage the MRDB which is structured according to the Vuel concept [Bédard and Bernier, 2002]. Without going into details, the Vuel structure is based on the multidimensional paradigm that governs most of today's decision-making applications [Inmon, 2002]. Multidimensional structures, as opposed to relational structures, are denormalized in order to provide fast response times from a data structure that better matches the way users see their data (i.e. by interrelated thematic dimensions having multiple levels of granularity). These structures rely on a fact table and on dimension tables in a star or snowflake schema [Thomsen, 2002; Giovinazzo, 2000]. In the Vuel structure, the fact table includes each combination of a geometric representation with a graphic representation and a semantic meaning, thus making a tuple from each vuel. The dimension tables include detailed information about the geometry, the graphic properties and the semantics of cartographic features as well as navigation paths between different levels of granularity of this information. Based on this paradigm, the Vuel structure allows the efficient management of three types of multiplicities (geometric, semantic and graphic) at the occurrence level. The complete description of the Vuel concept and its structure goes beyond the scope of this paper. Readers interested in more details are invited to consult [Bédard, Bernier, 2002].

This server makes data accessible through Internet via SQL queries (using a JDBC link). PostgreSQL[1], an open-source object-relational database, is used to manage descriptive data and the PostGIS[2] extension is used to manage geometric data according to the Simple Feature for SQL (SFS) specification [OGC, 1999].

[1] www.postgresql.org

[2] http://postgis.refractions.net

2.1.2. *Web Services*

The current version of UMapIT relies on two main web services; a first one intended to manage descriptive data and a second one specific to spatial data. The two services are deployed using Tomcat, an open-source servlet container that implements the Java Servlet and JavaServer Pages technologies developed by the Apache Software Foundation (http://jakarta.apache.org/tomcat/).

Descriptive Querying Service

This service offers an access to the descriptive data of the database using HTTP queries through Internet/Intranet. The results are returned to the client in a XML format. Through specific requests (defined by the services), it informs the Web client about the available data stored in the database. For the next versions of UMapIT, we would like to improve this component so it becomes a generic MRDB querying service (such as WFS for spatial data).

Spatial Querying Service

This service uses GeoServer, an open-source project that implements the WFS (Web Feature Service) standard [OGC, 2002]. It gives access to feature-level geospatial data through generic HTTP/XML queries. As opposed to traditional map servers, the client now obtains fine-grained information about geospatial features. This information is sent to the client in the GML (Geography Markup Language) format, which is an XML-based format for spatial data [OGC, 2003].

2.1.3. *Web Client*

The web client is used to create maps and interact with them (figure 2). It is developed in JAVA and based on Batik, an API (Application Programming Interface) that provide basic tools to visualize and interact with SVG and that can be customized and enriched using JAVA (xml.apache.org/batik). The spatial data (in GML) obtained from the spatial querying service are transformed into SVG (Scalable Vector Graphic) using XSL files before being displayed on the graphical user interface [W3C, 2001].

According to the following figure, the interface is divided in three major components: the selection module (left), the visualisation module (center) and the navigation module (right). The selection module is mainly composed of a semantic tree (1) while the GenZooms (2) and the Drills (3) are part of the navigation module. These functionalities are further explained in the following section.

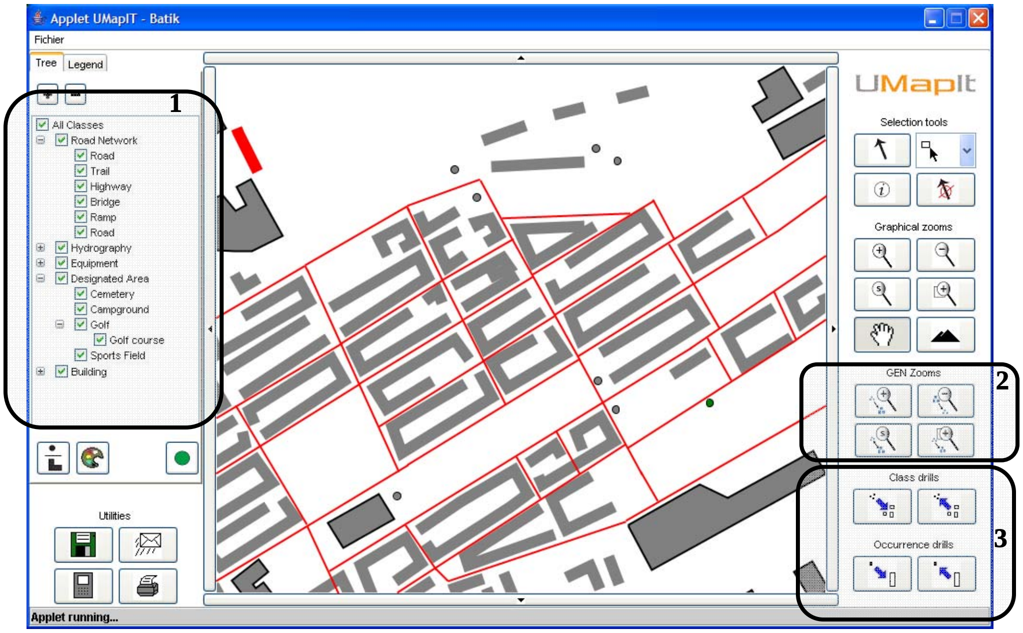

Figure 2 *UMapIT web client interface*

2.2. UMapIT Functionalities

As for every web mapping applications, UMapIT offers traditional navigation functionalities such as zooms, pan and fit. These functions are purely graphical as they only enlarge or reduce the map, move its centre point or display its full extent (using zooms capabilities). However, with a MR database, it is possible to go well beyond these functionalities and provide the user with more flexible and customizable possibilities. By properly using the multiplicities of its underlying database, UMapIT offers innovative operations such as drills and intelligent zooms.

Drills

The drill operations come from the On-Line Analytical Processing (OLAP) domain (Thomsen, 2002) and allow the navigation from one level of detail (LOD) to another. The operation that goes from a detailed LOD (e.g. large-scale map) to a simplified LOD (e.g. small-scale map) is called *drill-up* while the one going from a simplified (generalized) level to a detailed level is called *drill-down*. These drill operations can be used for a class or for a single occurrence.

Class drills

A class drill operation affects all the occurrences of a same object class. For example, suppose that the buildings are polygons at the 1K scale. A **class drill-up** operation will replace all the buildings with a simplified geometry. Though this operation affects all the occurrences of an object class, they are not affected the same way. The polygonal geometry of some of them may be replaced by a point geometry while other occurrences (usually the larger ones) may remain polygonal (but a simplified polygon). This depends on the next geometric level of each object. Although this is done using the MR links, the user perceives the drill-up operation as on-the-fly generalisation.

The **class drill-down** operation is used to increase the geometric LOD of the occurrences of an object class. This operation will display the occurrences using a more detailed geometry, which is stored in the following (below) geometric hierarchical level associated to each occurrence. The following figure illustrates a class-drill down on the Building class from the representation shown on figure 2. Each building has been replaced by its detailed geometry (when available).

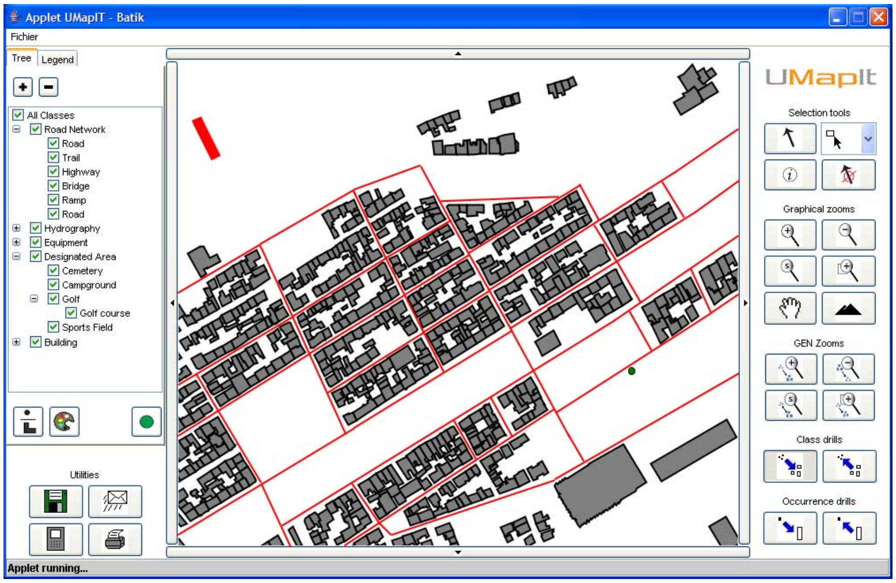

Figure 3 *Result of a class drill-down operation on the Building class*

This drill-down operation may also involve 1:n relationships when the Vuel structure includes aggregation relationships (as it is the case for the above example).

Occurrence drill

For some applications, it can be relevant to navigate among the different LODs of a specific object. For example, some objects may be used as landmarks and should be easily emphasized with additional details. Given that the spatial database of the UMapIT application stores the multiplicities at the occurrence level, drill operations can be applied on a unique object or a group of objects. As for the class-drill operation, the occurrence-drill may be used to represent an object with more or less detail (respectively drill-down or drill-up). Figure 4 shows the result of an occurrence drill-down (from the figure 2) on a specific object (the building in the right bottom corner).

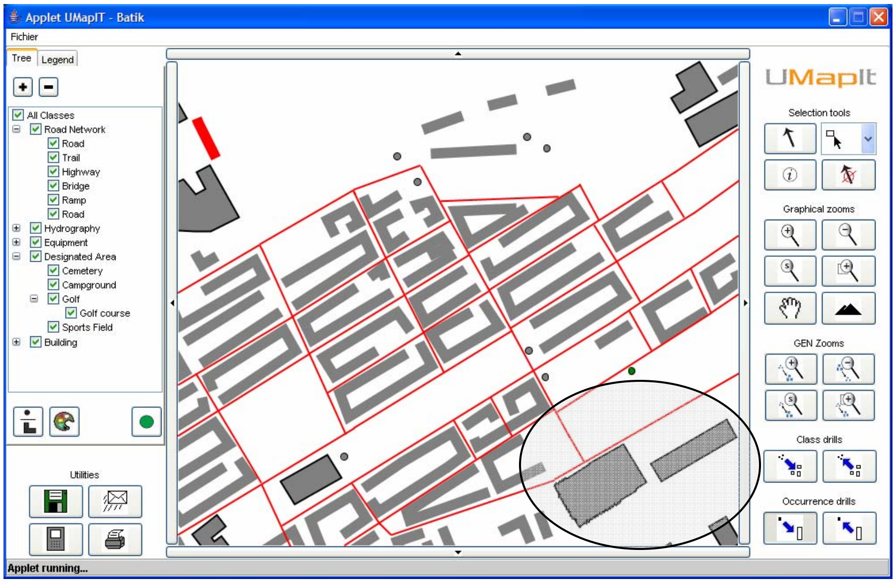

Figure 4 *Result of an occurrence drill-down*

GenZooms

By joining zoom operations with drill operations, we can provide *intelligent zoom* functions that adjust the content of a map depending on its display scale (Timpf & Devogele, 1997). For instance, a zoom-in operation could be coupled to a drill-down operation such as the enlarged resulting map shows more detailed objects. Accordingly, a zoom-out operation with drill-up capabilities would truly result in an on-the-fly generalisation. We call these special zooms GenZooms (*Generalisation Zooms*).

Our GenZooms are based on the occurrence level. This is a major difference compared with existing applications that offer layer-based zoom operations. In such operations, these applications replace the actual layer (e.g. building) with another one (e.g. urban area). Every occurrence is treated the same way. Usually, they just appear or disappear (selection/operation). With a MR management at the occurrence level, it is possible to take into consideration the specific characteristics of each object.

Actually, our GenZoom operation is based on the object size, on the screen resolution and on the scale. When it goes beyond a determined interval (e.g. 10K to 50K), the object is displayed using its simplified (zoom-out) or detailed (zoom-in) geometry. Accordingly, a GenZoom won’t necessarily affect all the occurrences of an object class. Some of them may be more or less simplified (given the LOD of the next geometry of the object) while others may remain unchanged.

3. Future functionalities

Semantic drills

At the time of writing this paper, the drill operations are only for the geometry of objects. However, the underlying MRDB database also supports semantic multiplicity. A same geometric representation may thus be associated to several semantics. These semantics may be at different LODs (e.g. Construction -> Building -> House) or may results from different points of view. The semantic drills would allow the user to navigate between the different semantics that can characterise a specific geometry. In another prototype, such functionality was already offered [Bernier 2002].

Geometric drills using semantic parameters

The actual geometric drills are solely based on explicit relations that link the different geometries of a same object. An occurrence drill-down results in a database query that searches for the next geometric representation of the selected object. However, it could be interesting to drive the geometric drills by semantic information. For example, from an existing map, one may whish to display public buildings with a more detailed geometry compared to private buildings.

Semantic and geometric drill-across

The drills that were discussed until now are hierarchical operations that are based on a vertical navigation (top-down or bottom-up). However, the multiplicities in a database are not always multiscale. They can results from different points of view, for the same scale [Vangenot, 2004]. For example, the UMapIT database was filled by integrating several data sources from different data producers. Two of them were having data for the 250K scale. Though the data were produced for the same scale, they were acquired using different rules which result in geometric and semantic divergences. The navigation between these geometric (or semantic) representations may be achieved using a drill-across operation.

Graphic aspects

Actually, the graphical aspect (semiology) is not fully exploited in UMapIT. The user can only define some graphic properties (fill color, line weight and style) in an ad-hoc manner. In the next version, we would like to strengthen this aspect by proposing some predefined styles to the user and allowing him to define his own. This could be achieved by storing different XSL files, each of them defining a specific complete semiology. The user would be able to select among these predefined styles, or define new ones.

Taking into consideration the graphic aspect of the data, it is possible to define new graphic-based operations. These operations allow the navigation from a graphic point of view. For example, a graphic-drill down on “Buildings” that are originally presented in blue could results in different hints of blue: dark blue for high-value buildings and light blue for low-value ones. This capability is very interesting when it comes to create thematic maps.

Define new geometries or semantics

In an attempt to develop more flexible and customizable applications, it would ultimately be necessary to allow the user to integrate his own geometries and/or semantics to the system. This can be viewed has a sort of update. For example, the newly geometry should be stored in the adequate LOD and links to the other representations of the object (if it exists in the database). From a semantic point of view, this means that the user would be able to integrate and use its own ontology. For example, two applications domains may not use the same vocabulary to define and describe objects, leading to semantic multiplicity.

Add Generalisation capabilities

Today's version of UMapIT entirely relies on multiple representations to simulate on-the-fly generalisation by efficiently exploiting the MR links. Though this approach offers fast response times, feeding and maintaining the database remains complex. To reduce this problem, we will add on-the-fly generalisation procedures in some circumstances using a new concept called SGO (Self-Generalising Object) that combines generalisation algorithms with geometric patterns at the occurrence level. SGO are special types of objects that are able to generalise a cartographic object using one or more (1) geometric patterns, (2) processing patterns and (3) spatial integrity constraints. These SGO are tagged to exact geometries to create MR with auto-generalisation capabilities and will be used during operations such as zooms and drills. SGO are being investigated in a PhD thesis [Sabo, 2004] while geometric patterns have already been defined [Cardenas, 2004].

4. Conclusion

This paper has presented an on-demand web mapping application called UMapIT. This application relies on a MRDB that is structured according to the VUEL concept. Having such an underlying database, it is possible to define new types of tools that benefit from the multiple representations stored in the database. In particular, these tools allow the user to easily navigate among the different levels of detail of the MRDB, from a geometric, semantic or graphic point of view and to simulate on-the-fly generalization.

UMapIT still needs a lot of improvements and is just a first step towards truly on-demand web mapping applications. From a technological point of view, we face some problems and limitations regarding the data transfer over the network and the on-the-fly transformation of GML data into SVG (using XSLT). These problems are mostly due to the large data volume. The actual version of UMapIT is based on multiple representations but this application will gain in flexibility and performance as on-the-fly generalisation mechanisms will be integrated. Advances in this research area are promising and will help to move towards truly on-demand mapping applications.

Acknowledgement

The authors wish to thank Canada GEOIDE Network of Centres of Excellence for funding this research with the following organizations: Natural Resources Canada, Research & Development Defence Canada, Natural Resources Quebec, Intergraph.

Bibliography

BASARANER, M., M. SELCUK, 2004, An attempt to Automated Generalisation of Buildings and Settlement Areas in Topographic Maps, ISPRS XXth Congress, Commission IV, WG IV/3, 12-23 July, Istanbul

BÉDARD, Y. & E. BERNIER, 2002, Supporting Multiple Representations with Spatial View Management and the Concept of "VUEL"., Joint Workshop on Multi-Scale Representations of Spatial Data, ISPRS WG IV/3, ICA Commission on Map Generalisation, 7-8 July, Ottawa

BERNIER, E, 2002, Utilisation de la représentation multiple comme support à la généralisation de vues de bases de données géospatiales dans un contexte SOLAP. M.Sc. thesis, Dept. Geomatics Sciences, Université Laval, Canada, 95 pages.

CARDENAS, A., 2004, Utilisation de patrons géométriques comme support à la généralisation automatique à la volée. M.Sc. thesis, Dept. Geomatics Sciences, Université Laval, Canada, 87 pages.

GIOVINAZZO, W.A. 2000. Object-Oriented Data Warehouse Design: A Star Schema. Prentice Hall, 349 p.

INMON, W.H, 2002. Building the Data Warehouse, 3rd Ed. John Wiley & Sons. 412 p.

OPEN GEOSPATIAL CONSORTIUM, 1999, Simple Features Specifications for SQL, v. 1.1, 78 p.

OPEN GEOSPATIAL CONSORTIUM, 2002, Web Feature Service Implementation Specification, v. 1.0.0, 105 p.

OPEN GEOSPATIAL CONSORTIUM, 2003, Geography Markup Language (GML) Implementation Specification, v. 3.0.0, 538p.

SABO, N.M. 2004. Intégration des algorithmes de généralisation et des patrons géométriques pour la création d'objets auto-généralisants afin d'améliorer la généralisation cartographique automatique. PhD research proposal, Dept. Geomatics Sciences, Université Laval, Canada, 62 pages.

THOMSEN, E., 2002, Olap Solutions : Building Multidimensional Information Systems, Second Edition, John Wiley & Sons, 688 p.

TIMPF, S. AND T. DEVOGELE, 1997, New Tools for Multiple Representations. ICC'97, Stockholm, Editor: Lars Ottoson, pp. 1381-1386.

VANGENOT, C., 2004, Multi-Representation in spatial databaes using the MADS conceptual model, ICA Workshop on Generalisation and Multiple Representation, Leicester, 20-21 August

W3C, WORLD WIDE WEB CONSORTIUM, 2001, Extensible Stylesheet Language (XSL), v.1.0, http://www.w3c.org/TR/xsl

W3C, WORLD WIDE WEB CONSORTIUM, 2003, Scalable Vector Graphics (SVG) 1.1 Specification, http://www.w3c.org/TR/SVG11/

WEIBEL R., E. BERNIER, Y. BÉDARD & A. CECCONI, 2002, La generalisation à la volée, in : Généralisation et Représentation multiple, A. Ruas (ed.), Hermes, p. 319-335.