ON-DEMAND WEB MAPPING: TOWARDS AN OCCURRENCE-BASED APPROACH

BÉDARD, BERNIER, LAMBERT

1. Introduction

Recent years have witnessed the emerging of web mapping applications. These applications allow a user to have access to spatial data, usually by means of predefined maps. Who have never used the well-known MapQuest[1] application to get driving directions? Although this kind of applications is very useful, it is rather limited because it is designed for well-defined purposes (road trip planner, driving directions, etc.). With such applications, users have very few options to create their maps. Usually, they can select the scale (among a pre-existing coarse set) and the themes to be displayed. They have no regards concerning the geometry, the semantic or the graphic characteristics of the object classes. Besides, there is a growing interest in developing more powerful web mapping applications by taking benefit from technologies such as GeoMedia Webmap. These technologies offer a great number of options to create maps but remain heavy and complex for mass users, hence the need to develop a new approach based on this kind of technologies that will allow users to create maps in an easy and intuitive way.

This paper presents a new web mapping application offering map-on-demand capabilities that uses Intergraph technologies. This application allows a user to create maps according to his specific needs in a very flexible way. The user can select the object classes to be displayed, as well as how they will be displayed (geometry and semiology). Moreover, this application offers very strong navigation capabilities as it supports multiple representations at the occurrence level (independently of map scale). Accordingly, the user can navigate among the different geometries of an object class as well as of a specific occurrence. Hence, as opposed to existing intelligent zooms offered in some applications where changing the scale imply that all objects of a same class behave similarly (change its geometry, appear or disappear), it is possible to change the geometry of a specific object without modifying the geometry of all objects of the same class.

This map-on-demand application is developed in the context of the GEMURE project (Generalization and Multiple Representations for On-Demand Map Production and Delivery), a research project funded by the Canadian Network of Centres of Excellence in Geomatics - GEOIDE.

2. The underlying structure of the map-on-demand application

In order to provide the user with very flexible means when creating a map, we must have a flexible database view engine that supports simultaneously geometric multiplicity, semantic multiplicity and graphical multiplicity. The three types of multiplicity are necessary to fully harness the power of spatial databases and to fulfill the needs of map-on-demand users.

Geometric multiplicity refers to an object that can be represented with different geometries. This is normally the case when the same object can be displayed at different scales. For example, a building would be represented with a detailed polygon at a large scale and with a simplified polygon or even a point at a smaller scale (figure 1). However, we can also have different geometries for a same scale. Nowadays, organizations tend to integrate all their data (coming from disparate, heterogeneous systems) in a unique data warehouse. This data warehouse serves many different applications and each of them has its own needs regarding how an object has to be displayed.

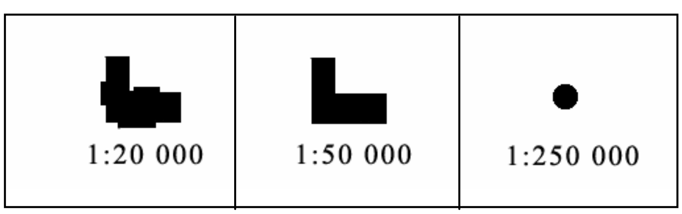


Figure 1 *Multi-scale geometric multiplicity*

[1] www.mapquest.com

For example, some applications need to display houses according to their roofing while other applications might display them according to their foundations (figure 2). We refer to this situation as a “single-scale geometric multiplicity”.

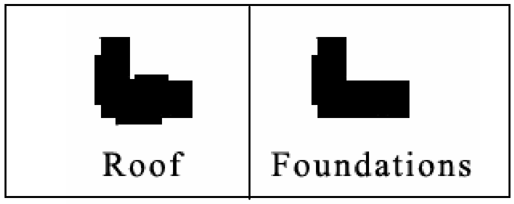
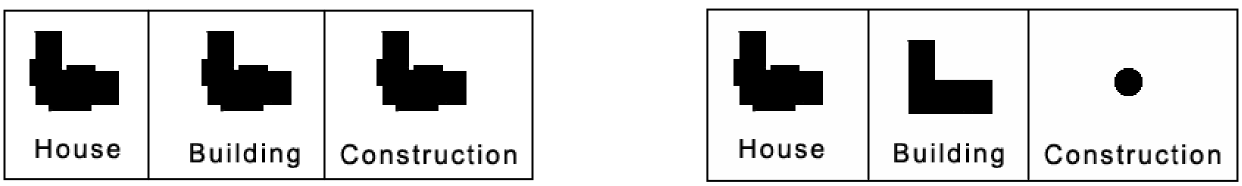


Figure 2 *Single-scale geometric multiplicity*

Semantic multiplicity arises when a same cartographic element can be associated with different definitions (figure 3a). Once again, these semantics can differ according to the scale or to the application. For instance, at different scales, a same cartographic element can be defined as a house, a building or a construction. Also, given that each application domain has its own ontology, a same cartographic element could have different meanings, regardless of the scale. Furthermore, one can mix semantic multiplicity with geometric multiplicity as shown in figure 3 b).



a) b)

Figure 3 *Semantic multiplicity a) for a same geometry, b) for different geometries (depending of the scale)*

Finally, the **graphic multiplicity** occurs when a same cartographic element is represented by different visual variables following different semiology rules. As for the two other kinds of multiplicity, this one can be associated to a change in the display scale. For example, the symbols used to represent certain types of elements can be different from one scale to another or from one application domain to another (figure 4)

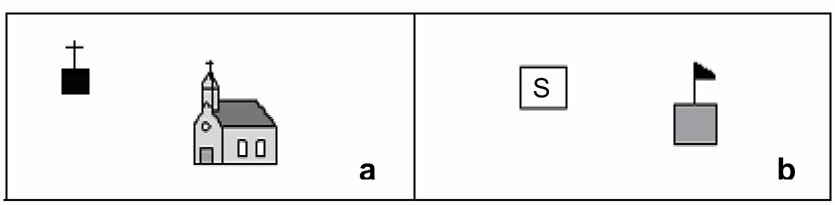


Figure 4 *Graphic multiplicity a) multi-scale b) single-scale*

In order to support these three kinds of multiplicity in spatial databases, we’ve developed a new concept called VUEL, for View Element. A vuel is defined as the basic construct of a spatial database view, as the pixel (picture element) is the basic element of an image. A vuel represents any element that can appear on a spatial database view [Bédard et al., 2000]. The vuel is in fact, a unique combination of a visible element (geometry and graphic semiology) with a given semantic [Bernier, 2002]. Accordingly, the vuel is not the semantic object in itself (e.g. house) but what we can see in a view, for example a hatched blue polygon with house attributes. The same object of the reality can also be represented by a red point in another view or it can becomes spotted green and defined as a building with different attributes and so on (figure 5).

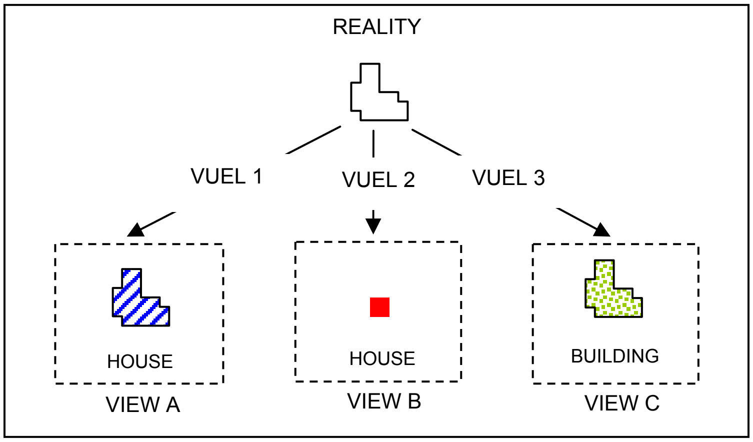


Figure 5 Three vuels used to define or represent a same real world element

This structure is well suited for web mapping applications offering map-on-demand capabilities that require fast answer times because vuels are organized in a multidimensional data structure (also called *Hypercube* in datawarehousing vocabulary) where most data are stored explicitly and aggregation/generalization algorithms used only when possible. The implementation structure is beyond the scope of the present paper and won't be discussed here. The reader is invited to consult [Bédard & Bernier, 2002] for more details.

3. The map-on-demand application

Based on the VUEL concept, our map-on-demand application is very flexible and allows a user to create a map upon his own constraints in an intuitive way given that no query language has to be used. All the map creation process and the navigation are done only with some mouse clicks. The user doesn't even have to touch his keyboard.

For that application, our data has been stored in an Access Warehouse via GeoMedia Professional. The interface has been developed in ASP and the cartographic part is based on GeoMedia WebMap (figure 6).

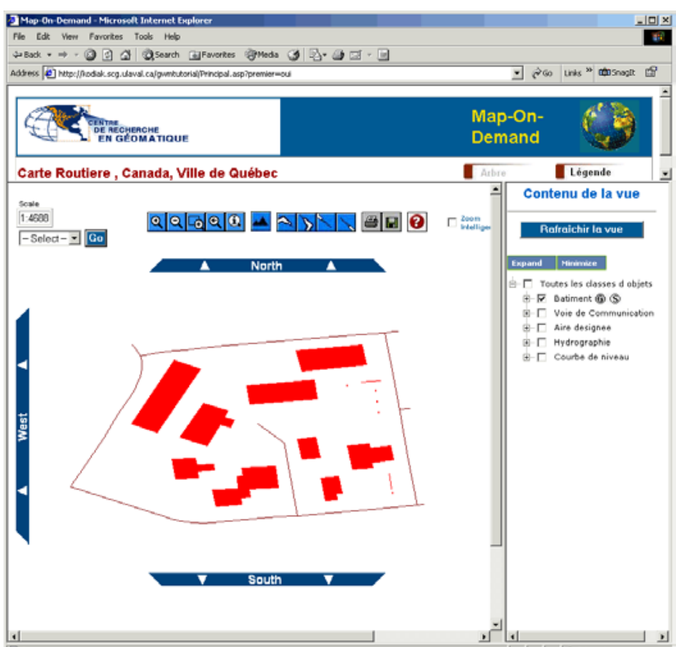


Figure 6 The interface of the map-on-demand application

The VUEL Tree

At first, the user selects the object classes that he wants to see. This is done using the “VUEL” tree (figure 7). This tree presents the available object classes at different levels of semantic granularity. Accordingly, the user can select all the buildings or he can be more specific in selecting only those of interest for him such as “Houses” or “Apartments“. Once the selection is done, the objects are displayed in the cartographic part.

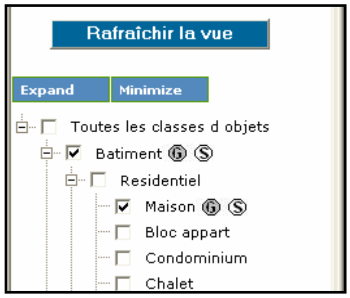


Figure 7. The VUEL Tree

Because the underlying structure supports geometric and graphic multiplicities, we can store predefined semiology profiles that govern the way objects are displayed. For example, semiology rules for urban development plans differ from the ones used in topographic maps. Also, the user has the possibility to create new semiology profiles very easily. For instance, for each object class that has been selected, the user can change the geometry (going from a general geometry to a detailed one) or the graphic characteristics using the “G” and “S” buttons beside the class name (figure 7). For example, the user may wants to change the color of an object class or of a specific object (figures 8, 9). It is also possible to change the line style to be used for linear elements (figure 10).

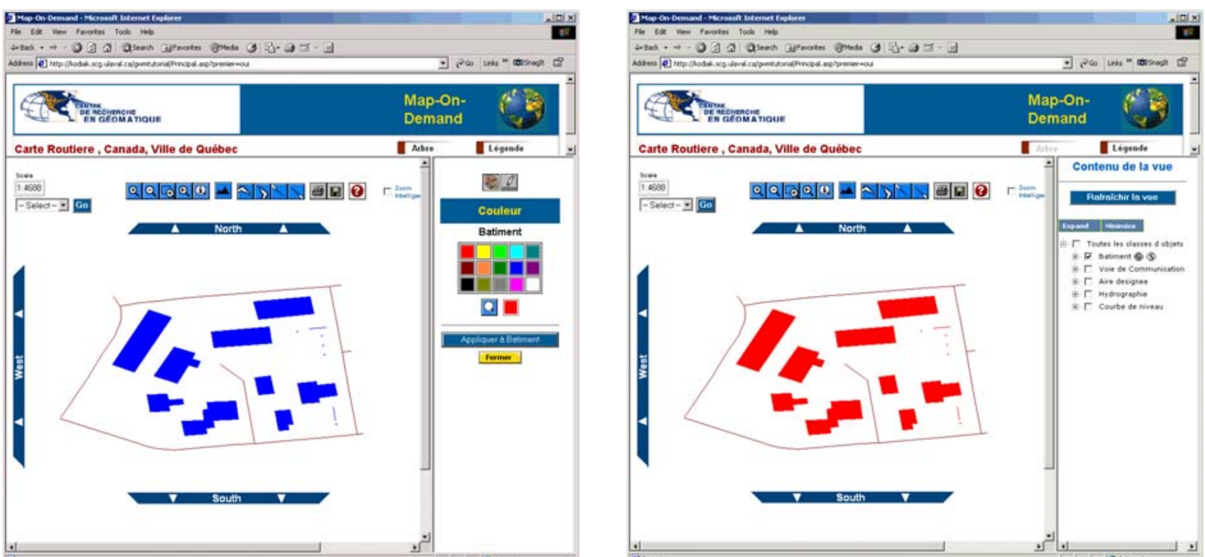


Figure 8. Changing the color of an object class

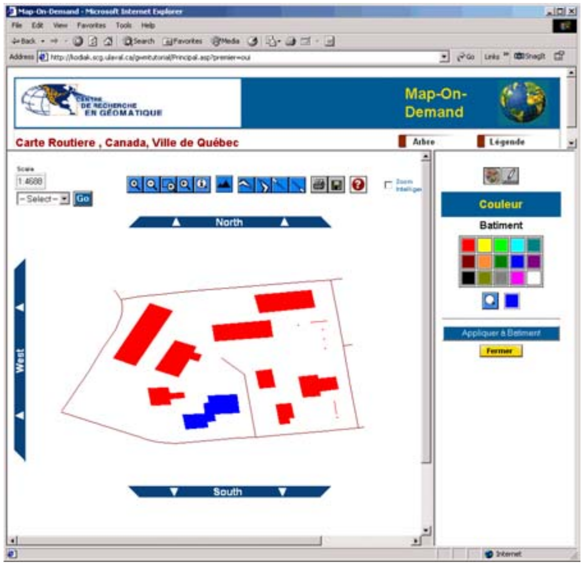


Figure 9. Changing the color of a specific object

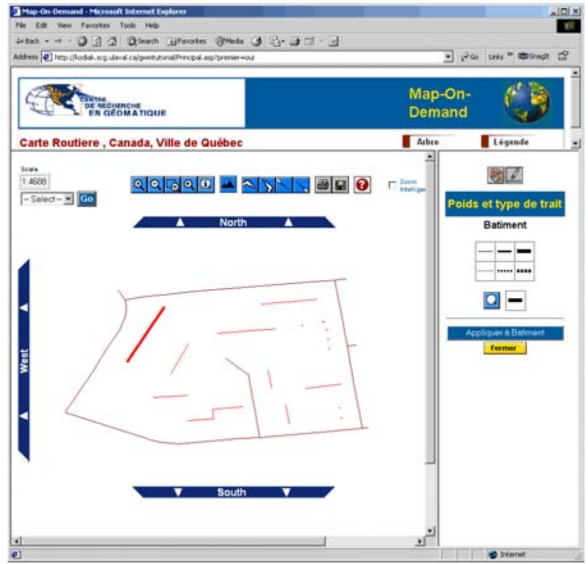


Figure 10. Changing the line thickness of a specific object

Navigation functions

The application provides the users with some functions to easily navigate among the spatial data in order to create different views. On one hand, there are the traditional functions such as zoom in, zoom out, zoom to box, previous and fit (figure 11). On the other hand, there are some functions that allow the user to select the most appropriate level of details (depending on his own needs) for representing or defining an object or an object class. These functions are called Drill down and Drill up and allow the user to navigate among the different granularities associated to elements, on a geometric, semantic or graphic point of view.

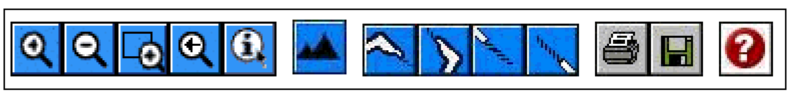


Figure 11 Spatial navigation functions

For example, at first buildings may appear as points. Using the *Class Drill-down* tool, the user can get the linear geometries for all objects in the Building Class (figure 12).

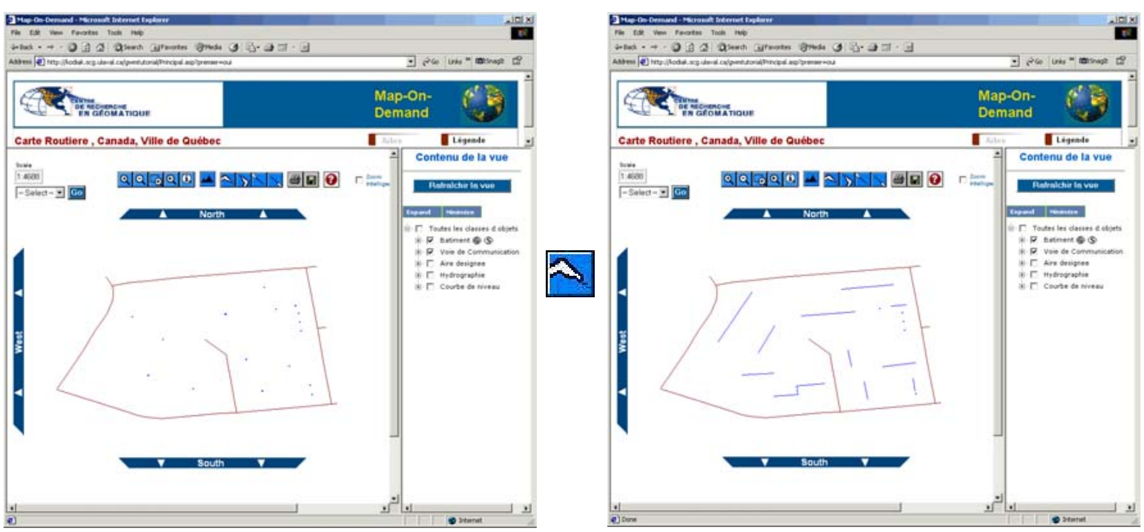


Figure 12. Drill-down on the Building class

Also, if the user wants to point out at a specific element (e.g. his house), he can drill-down on the geometry of the specific object with the O*bject Drill-down* tool (figure 13) which results in displaying a more detailed geometry for the selected object.

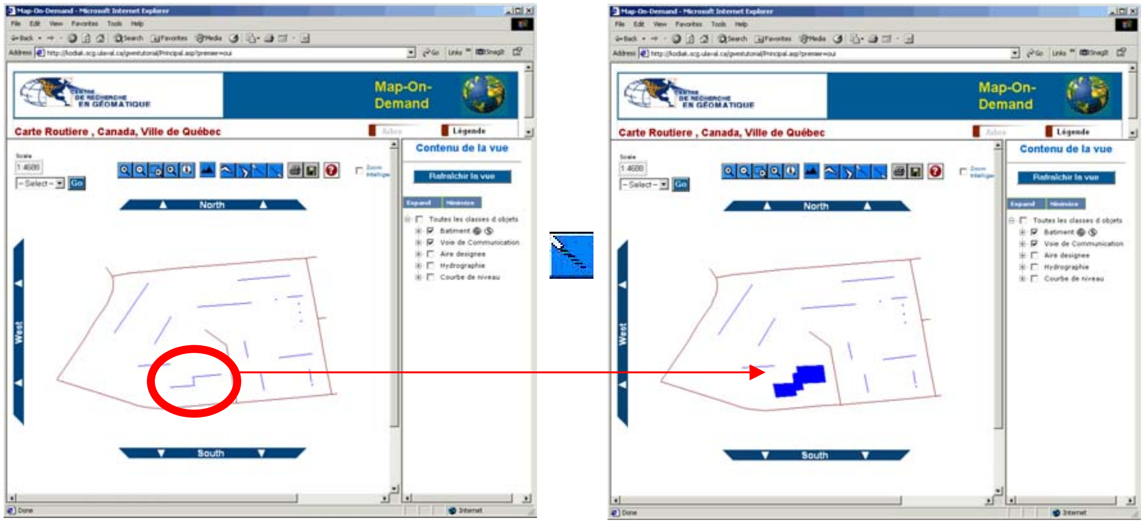


Figure 13 Drill-down on a specific object

These drill-down or drill-up operations can also be used on descriptive data. Hence, one can drill-down on a specific attribute to have more details about this attribute. For example, a drill-down operation on the "Commercial" value for the "Utilization" attribute of a building will result in displaying a more specific value such as "Restaurant".

4. Conclusion

In order to have powerful map-on-demand applications, we must rely on a very flexible database view engine. The VUEL structure, because its support geometric, semantic and graphic multiplicities, is well suited for this kind of application. Based on that structure, this paper has presented a web mapping application where the user has all the flexibility to create maps that will satisfy his own requirements in a very fast and intuitive manner. This application still needs to be improved but we believe that is a first step towards a new generation of web mapping applications, offering map-on-demand capabilities.

5. References

Bédard Y & Bernier E, 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 Com. on Map Generalization. Ottawa, CANADA, July 7th-8th.

Bernier E., 2002, "Utilisation de la représentation multiple comme support à la génération de vues de bases de données géospatiales dans un contexte SOLAP", Mémoire de M.Sc., Faculté de foresterie et de géomatique, Université Laval, 95p.