

A web application for landslide inventory using data-driven SVG

Maurizio Latini, Barend Köbben

latini@unisi.it – Centro di Geotecnologie - Università di Siena, Via dei vetri vecchi 34, 52027 San Giovanni Valdarno, Italy;

kobben@itc.nl – International Institute for Geo-information Science and Earth Observation (ITC), PO box 6, 7500AA Enschede, The Netherlands.

Abstract: The landslide map in the Serchio basin (Central Tuscany, Italy) is an official document that represent the actual state of the landslides in the region. At present, the updating of the map is carried out by different municipalities using paper sketch maps, sent by post to the Autorità di Bacino del Fiume Serchio (AdB). The objective of the work presented here was to significantly speed up and simplify this updating process, while taking into account the severe constraints of the municipalities. This was to be achieved by providing them with a lightweight Web based map application that allows inventory of new landslides and submitting them via the WWW directly to the AdB databases. Open Source technology and Open Standards were employed to build a database-driven application. The data is stored in a spatial database backend (MySQL), following OpenGIS Simple Features specifications. Server applications extract data from the database and deliver it as a client-side application in SVG, the Scalable Vector Graphics format of the W3C. This paper presents the technical background and setup of this application and plans for future development.

The official landslide map of the Serchio basin

The process of updating geological and geomorphological databases is not always a straightforward operation. In Italy for example, landslide databases are in some cases the responsibility of central authorities, such as Regions, in others of local authorities, the so-called “Autorità di Bacino”.

This article is focussed on one of these, the *Autorità di Bacino del Fiume Serchio* (AdB), responsible for the Serchio basin, located in a zone close to the Apuane Alps in the North West part of Tuscany, Italy.

The landslide map of the Serchio basin is an official document that should represent the actual state of landslides in the region. Nowadays this map is updated every year, but this particular region is regularly affected by landslide phenomena, and a yearly updating sometimes is not sufficient.

The current updating process

The current updating system, shown in figure 1, is depending much on the local municipalities in the region. They usually receive advice of new landslide phenomena from citizens. The location of these phenomena is sketched on a printed map, which is submitted to the GIS department of the AdB via surface mail. At the AdB offices, an operator has to draw new polygons representing the new landslide phenomena on a temporary layer of the landslide GIS. Before updating the official landslide map with the occurrence of the new phenomena, a field check is needed, for which the sketch layer is printed and team consisting of a geologist and a geomorphologist is sent to the field.

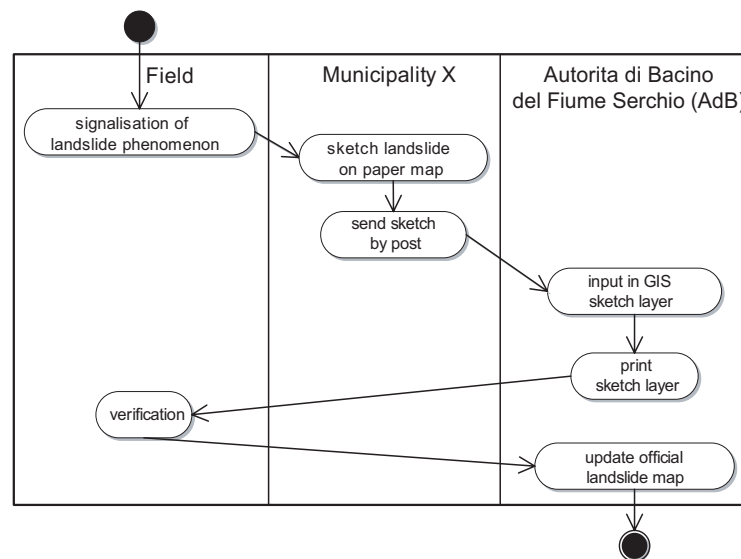


Fig. 1. Current landslide map updating process

Their objective is to decide if the landslide on the sketch map represents a possible damage for the population. If the phenomena can really affect

either infrastructures or private properties the official landslide map is updated with the appropriate polygons.

There are many factors that slow down the updating process and make it less efficient, especially the cumbersome sketching and posting of these sketches to AdB where they subsequently have to be digitized.

Towards a more efficient updating process

The objective of the work presented here was to significantly speed up and simplify the updating process, while taking into account the limited possibilities of the municipalities. Many of these are small and located in inaccessible sites. They also lack the staff and funds necessary to implement and maintain complicated ICT systems. Another problem due to their location is the lack of a fast internet connection. If available, the connection to the web is usually made via telephone line and a 56K modem. All these circumstances dictate that they cannot be expected to use a full-blown Geographical Information Systems.

Making the process more efficient, while at the same time coping with the restraints, had to be achieved by providing the municipalities with a lightweight mapping application running in a Web-browser, that allows the recording of new landslides directly to the AdB database. The map application had to have some basic navigation functionalities, such as zooming, panning and layer and legend control. Beside that, it should offer also a “digitizing” tool that allows the user to draw new landslide features which, once digitized, are submitted via the WWW directly to the AdB GIS. As an added bonus, the same system could be used by the AdB fieldwork teams to digitally finalise the official landslide map, as shown in figure 2.

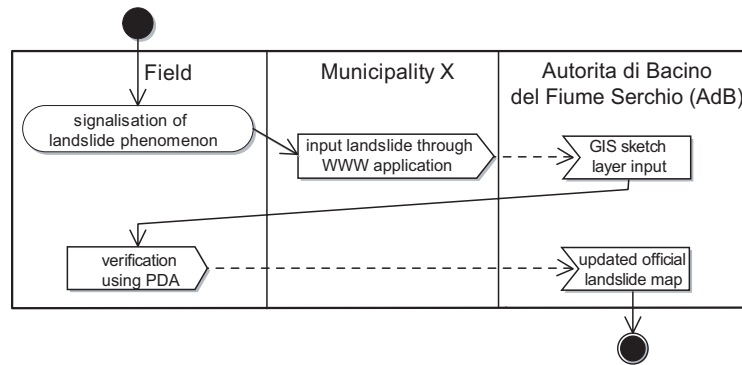


Fig. 2. The proposed future setup of the updating process.

In order to achieve the goals described above, Open Source technology and Open Standards were employed to build a database-driven application suitable for use on many platforms. The setup of this system was based upon ideas on a “GDI-light” architecture, being developed at the International Institute for Geo-Information Science and Earth Observation (ITC).

GDI^{LIGHT} – lightweight Geo-Data Infrastructures based on Open Standards and Open Source software

The term GDI might be usually connected with (very) large regional or national spatial data warehouses, but it is defined more generally as “the networked geospatial databases and data handling facilities, the complex of institutional, organizational, technological, human and economic resources (...) facilitating the sharing, access to, and responsible use of geospatial data at an affordable cost for a specific application domain or enterprise” (Groot 2000). In many cases GDI data and application infrastructures are being developed using high-end geospatial software solutions and large corporate databases, needing substantial investments in financial and human resources. But the principles of GDIs can be applied in simpler and more cost-effective ways just as well, which is of particular interest for the students and partners of the International Institute for Geo-Information Science and Earth Observation (ITC). ITC is an institute that aims at capacity building and institutional development specifically in developing countries (URL 1).

GDI^{LIGHT} (formerly called GDI-EMERGE) is an internal ITC project to employ webservices and a data back end to build light-weight, low-cost Geo-Data Infrastructures, using Open Standards and Open Source software. It serves as a general purpose testbed for applied as well as fundamental research activities, and should provide researchers and students alike with a proof-of-concept platform for relatively simple, low-cost, yet powerful ways of sharing data amongst various distributed offices and institutions as well as the general public. It is not the intention that it should grow out to be a fully working, coherent system, but should be seen as a *testbed* in the broad sense of “equipment for testing”. It is the place where we can show fellow researchers, consultants and students as well as possible users (such as GIS users from developing countries) that the things we teach can be made to work quite quickly, in a relatively simple and low-cost setup. The main building blocks of the system are:

- A spatial database back end that stores the geometry and the attribute data; Spatial data should be stored using the OpenGIS (OGIS) Simple Features specifications. In GDI^{LIGHT}, MySQL is used at present.

- A set of interoperable webapplications that interface with the database and with each other, and fulfil tasks such as delivering data in SVG for visualisation purposes and other XML formats (such as GML) for data exchange, serve data in OGC formats to and from the database using web services, etcetera. In GDI^{LIGHT}, these are developed using Java server technology, and at present deployed using the Open Source Apache Tomcat server.
- Simple Web-based interfaces enabling access to the maps and data for both desktop browsers and mobile platforms, as well as more sophisticated interfaces, for example providing data through an OpenGIS Web Feature Server to GIS clients. At present, we concentrate on web-browser clients, using SVG for graphics and interaction.

In figure 3, the conceptual setup of the system, such as it is currently being implemented, is shown. We do intend this to be a flexible setup which is expected to change over time. The current work ongoing is focussed on the processes represented by the darker arrows.

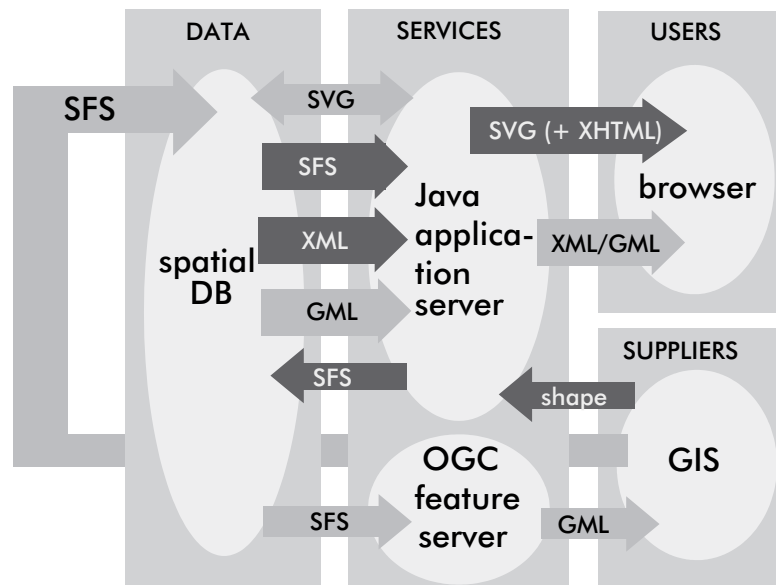


Fig. 3. Conceptual setup of GDI^{LIGHT}. Darkest arrows show current focus (SFS stands for OGC's Simple Features Standard).

A first application of the GDI^{LIGHT} testbed called *RIMapper* (Köbben 2004) was developed in early 2004 to look into the possibilities of generating light-weight, versatile Risk Indicator Maps (RIMs) from online data-

bases. RIMapper uses a JSP Tomcat server application to extract OpenGIS Simple Features stored in MySQL Spatial Extension and delivers these to mobile web clients as interactive SVG maps, based on XML configuration files. These maps are to be part of an urban risk management system, and therefore needed to fit a multitude of use cases, ranging from providing the general public with information about risks to providing local authorities an interface to the underlying risk assessment databases and models. Furthermore, the maps needed to be usable on a wide range of platforms, from the office systems of the local authorities to hand-held devices providing location based services to field personnel. Many of the techniques used in RIMapper could be deployed in the landslide inventory application that is the subject of this paper.

The spatial database

As the project set out to comply to open standards, there was the need to use a database that supported the standards of the OpenGeospatial Consortium (OGC, see URL 2). Among other things, the OGC has set the Simple Features SQL Specification that provides for publishing, storage, access, and simple operations on spatial features (point, line, polygon, multi-point, etcetera) through an SQL interface.

There are several databases with OGC-compliant spatial extensions, of which Oracle is probably the most prominent, but as another goal was to use open source software, two databases were under consideration: *PostgreSQL* (URL 3) a database system with *PostGIS* as an spatial extension and *MySQL* (URL 4). *MySQL*'s recent versions include OGC-compliant spatial extensions, although not implementing the full set of OGC specifications. The reasons for the choice of *MySQL* were the native Windows support and the simple 'lightweight' character of the software, as compared to the complicated though more fully-featured *PostGIS*. By adhering strictly to the OGC standards it should be straightforward to change or even mix database platforms in the future.

Geometric features are stored as OGC Simple Features geometry and transformed to other formats at run-time, by the application tier. The non-spatial attributes of the features are stored per layer in specific tables ('homes', 'roads', etcetera), related with the spatial features through their ID. A 'layers' table is provided as a per layer link to a 'styles' table, for layers that should be styled uniformly, eg. all roads sharing the same visualisation. Whenever the visualisation should depend on some data attribute per feature, eg. for a chorochromatic map of homes viewed by vulnerability type, the link is made from the data specific attribute table directly to the 'styles' table. The choice for the visualisation type mentioned above is

directed by the XML map configuration file (see next paragraph). A further two tables, ‘fragments’ and ‘actions’ are also for use by the application tier, storing SVG code fragments and ECMAScript event listeners, respectively.

The application tier

The heart of the RIMapper system are the services provided by a set of Java servlets. In our case, the application tier runs on Tomcat, a well-known open source servlet container from the Apache Software Foundation (URL 5), but the applications should run on any standards-compliant servlet container. The RIMapper system uses a set of generic Java classes to do recurring tasks like extracting OGC features and attribute data from the database, translating these into fragments of SVG and ECMAScript, collecting and structuring these fragments into valid output and delivering this output to the clients.

The glue provided to make all these parts act together are *XML map configurations*. They are parsed to get a description of the map needed and all its component parts. Below is a sample of an XML configuration for a very simple map of roads, which when clicked will show their ‘type’ attribute in a pop-up message.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE RIM PUBLIC "" "/RIMapper/RIM.dtd">
<RIM TYPE="SVG_STANDALONE" DB="rimapper" UN="X" PW="X">
  <TITLE>Clickable Data...</TITLE>
  <AUTHOR>ITC</AUTHOR>
  <HEADER>
    <FRAGMENT DBID="default" NAME="defSVGRoot"
      TYPE="SVG_ROOT"/>
    <STYLES>
      <STYLE DBID="default" NAME="defLine" TYPE="CSS"/>
    </STYLES>
    <FRAGMENT DBID="default" NAME="defInitPlusRIMmessage"
      TYPE="ECMASCRIPT"/>
    <FRAGMENT DBID="default" NAME="showRIMData"
      TYPE="ECMASCRIPT"/>
  </HEADER>
  <LAYERS>
    <LAYER DBID="default" NAME="roads" STYLETYPE="single"
      STYLE="defArea" ATTRIBS="type">
      <ACTION TYPE="simple" NAME="showRIMData"
        SCOPE="feature" EVENT="onclick"
        PARAMS="evt, 'rim', 'type'"/>
    </LAYER>
  </LAYERS>
</RIM>
```

In the header section of the XML map configuration, several fragments of SVG code are loaded to define gradients, symbols, filters and ECMAScript fragments to provide interactivity and other functionality. The last part, the footer section, simply declares the closing part of the output. In between is the body section that lists the actual layers of information to be mapped, used to retrieve the attributes and the geometry from the database. Layers have *types* that determine the way they are visualised cartographically, can have one or more *actions* that set the interactivity and *events* that set which event will trigger the action.

When all data needed has been collected by the system, the SVG output is composed and handed over to the web server for delivery to the client. The SVG generated will adhere to the SVG-Basic profile and will be suited for a broad range of clients, including PDA's.

Using this system, one can very flexibly offer database-driven maps on the web that are generated on the fly from the most recent data, and that can incorporate all the functionality, scalability and graphics quality that the SVG standard offers.

Implementation of the landslide inventory tool

Based on the needs of the municipalities described in the first paragraphs, the landslide inventory tool has to respond to some pre-specified characteristics. The steps needed to implement such a tool are:

- Creation of an SVG web map application using open source technology (MySQL, SVG) – based on the existing RIMapper system mentioned above.
- This SVG application should have basic functionalities necessary for the use as a landslide inventory tool (i.e. legend that can turn on/off the layers, identify features)
- Creation of an edit tool that will allow drawing a polygon in a sketch layer directly from the SVG application.

The map layout has been developed as shown in figure 4. The main map is located in the centre of the web browser/screen and is the only part of the screen that can be zoomed and panned.

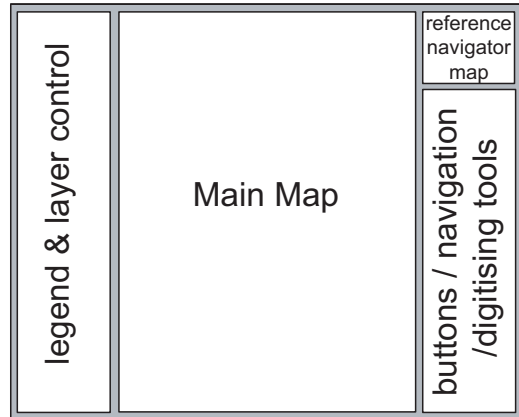


Fig. 4. Map Application Layout.

The other elements of the map application are a legend, a reference map for the navigation through the main map and a navigation tool where all the buttons and tools are located. At the time of writing the following functionalities have been developed:

Layers control via legend

A layer control is necessary in order to manipulate the main map. The map layers can be turned on and off like in a traditional GIS systems. When the map is loaded all the layers are set to visible, but the legend is invisible. To turn the layers off it is necessary to make the legend visible by clicking the button ‘Show Legend’ in the navigation tool. After this it is possible to set the visibility of the layers by simply using the check boxes next to the layer names. When the legend is turned off and then turned on again, the visibility state of the layers will be remembered.

Panning and Zooming

Due to the restrictions of the screen dimensions, map navigation tools are one of the most important functionalities for a screen map. As suggested by its name any SVG-viewer is ‘pan and zoomable’ by using key shortcuts. In this application however this intrinsic functionality has been turned off by setting the attribute ‘ZoomAndPan’ of the <svg> tag element to disable. Panning and zooming functionalities in this case have been obtained via calculations on the viewBox attribute of the svg file and by nesting <svg> elements. This approach was used, so that only the map is scalable and the buttons and the legend remain fixed on the side.

A series of buttons with arrows have been created in the Navigation Tool, by clicking on one of the buttons the map will scroll in the direction of the arrow. The central button is the Full Extent one, by clicking this button the map will come back to its original extent. Clicking on one of the zoom buttons will enlarge or decrease the details of the map view.

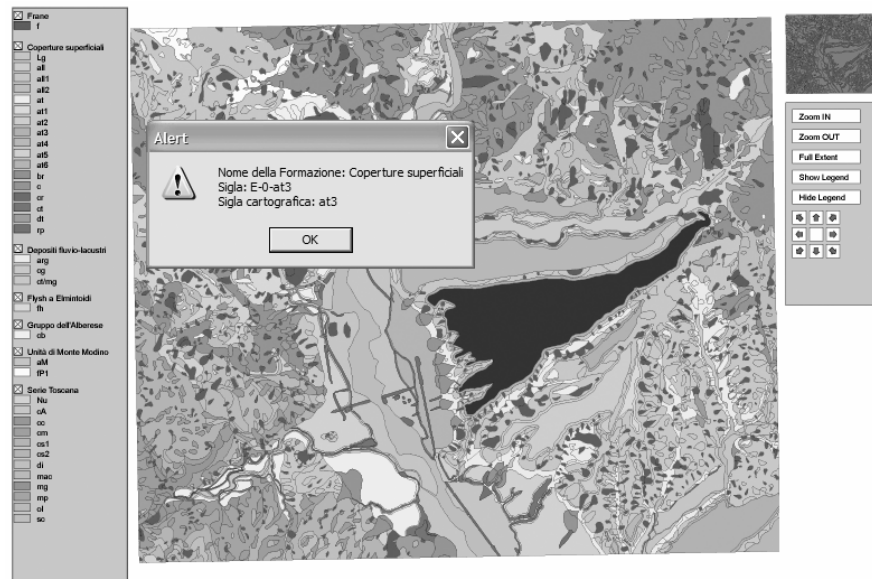


Fig. 5. Screen dump of the application, showing an information message box.

Interactive Overview Map

The interactive overview map is a good solution in order to navigate through a map. The overview map shows the whole map extent and it is also useful for an overall orientation. The overview map implemented contains a semi transparent rectangle which represents the corresponding view in the main map. The interactivity is achieved by adding a dragging functionality to the above mentioned rectangle. This will become a control by adding mouse events, if the user holds down the button and moves the rectangle around, a mousemove event is triggered. When the mouseup event is called the rectangle stays and the main map is centered to its new extent.

Map Layer information

When the mouse cursor is over the map the layers are highlighted, by clicking the layer of interest, a message box will pop up, as can be seen in figure 5. This message box contains the information regarding the clicked

feature like the name of the geological formation, the cartographic reference used and the rendering employed.

Showing coordinates

When the mouse cursor is over the map, in the bottom part of the control panel the map coordinates are shown. The calculation of the actual coordinates is really important, because the digitizing tool is based on this coordinates. It is also important because in a future development of the application it should be possible drawing the new polygons by entering its coordinates collected directly in the field with a GPS system.

Digitizing function

Once the map coordinates are calculated, implementing a digitizing functionality for the map was quite straightforward, by using the DOM-functionalities of the SVG map. In SVG, being an XML language, all elements are represented in a Document Object Model or DOM. This *DOM-tree* is exposed through API's to the viewer application. Therefore by scripting one can use dynamic arrays that store the X and Y coordinates for every mouse click. After the dimensioning and populating of the arrays the next step is to create a DOM node element (in our case a path element), that has to store the drawing coordinates and represents them as a polygon.

Before the actual drawing of the polygon a series of points and lines shows the proceeding of the digitizing process. The digitizing process is strictly related to the coordinate system of the map, whenever the map extent changes (because of zooming or panning) the map coordinates have to be recalculated. The steps of the digitizing process are (see figure 6):

- When the *Start Digitizing* button is pushed a message pops up
- During the digitizing the points are added together with the lines
- When the *Stop Digitizing* button is pushed the polygon is closed and displayed in blue colour.

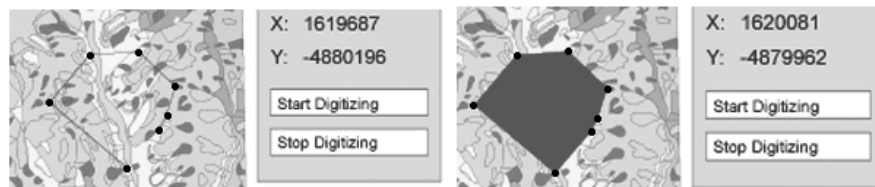


Fig. 6. Screenshots of the digitizing process.

Some refinement in the digitizing process should be made, by allowing vertices to be edited. At the moment, the newly digitized polygon is not yet uploaded to the AdB database. The client-side functionality needed for this (the *postURL* function) is already well-established in the current viewers, and will become an official standard soon. The only extra functionality needed is converting back the SVG to OpenGIS Simple Features, this could best be achieved server-side. However, the AdB still will need to rework the server-side to actually receive the inputs and store them in a safe, transactionally sound way.

An overview of the currently implemented and planned features can be seen in table 1.

Table 1. Implemented and planned functionalities for the landslide SVG Map.

Feature	Implemented	Planned
<i>Navigation:</i>		
Interactive Overview Map	X	
Panning	X	
Zooming	X	
Show coordinates on mouse move	X	
<i>Map controls:</i>		
Layer control	X	
Identify features	X	
<i>Digitizing operations:</i>		
Digitizing	X	
Move vertex		X
Delete vertex		X
<i>Client/Server Communications:</i>		
Update sketch layer with digitized polygon(s)		X

Conclusion and future work

The application described in this paper will be used by the municipalities that do not have a GIS system but at the same time need to update a sketch layer with new landslide phenomena. Via the WWW this authorities can access the official landslide map and, based on the new landslide phenomena occurrence draw a sketch polygon on the map. This sketch polygon can be later on submitted (always via WWW) to the central authority responsible for the updating. Based on these sketch drawings, experts will perform a field check and if the landslide phenomena results potentially dangerous for the population, the official landslide map will be updated.

The first step in the application developing has been the creation of a SVG map from an online database, based on the data-driven GDI^{LIGHT} sys-

tem. The choice of SVG has been made because it is an open source standard and through the use of ECMAScript highly customizable. Furthermore it permits to generate a light weight map that represents what these authorities needs.

The work described here is only the starting point of the Landslide Web Map. Future developments for this project will need to obtain a lighter SVG application, in fact the actual dimension of the test project are too big. The problem is that when the file is loaded all the geological formations are loaded in it; this is the main cause of the dimension of the file. In the future, a file loader will be developed. With it the user will be able to load only the data needed directly from the server. This will speed up the process because only a skeleton map will be loaded at the beginning.

Future developments will include also a more appropriate rendering for the geological formations, using the cartographic standards available for geological mapping. This rendering should include the representation of oriented symbols.

The digitizing tool also needs to be implemented with additional functionalities like the possibility to edit polygons and individual vertices. Another planned functionality will be the possibility to add attributes to the digitized polygon; this will be obtained via a popup window that the user will fill with the appropriate information.

With these functionalities added, a well-designed and useful digitizing tool should finally be achieved.

URLs

- [1] ITC web site: see <http://www.itc.nl/>
- [2] OGC web site: <http://www.opengis.org/>
- [3] PostGIS site: <http://postgis.refrations.net/>
- [4] MySQL AB site: <http://www.mysql.com/>
- [5] Apache Foundation Tomcat pages: <http://jakarta.apache.org/>

References

- Groot, R., McLaughlin, J. (2000): Geospatial data infrastructure – concepts, cases and good practice. Oxford University Press, Oxford etc.
- Köbben, Barend (2004): RIMapper - a test bed for online Risk Indicator Maps using data-driven SVG visualisation. In: Georg Gartner (ed.) Proceedings of Location Based Services and TeleCartography (Geowissenschaftliche Mitteilungen, Heft 66). Wien: Institute of Cartography and Geo-Media Techniques & ICA, pp. 189-195. <http://kartoweb.itc.nl/RIMapper/>