Geo-webservices for animated mapping of flood risks

Barend Köbben

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

Abstract: The goal of this paper is to present the cartographic possibilities for visualizing *time series* data of flood risks and the advantages of using *geo-webservices* to present these visualisations. First, the concept of animated maps is explained. Strengths and weaknesses of various types of animated data visualisation are presented. Secondly, the basic concept of geo-webservices and the importance of interoperability through the use of open standards is set out. Then we will look into the possibilities of combining these two recent developments to achieve *animating time in geo-webservices*: We will give an outlook on how the technology can allow animated information on the web, that retains links with the underlying data and data models, so that a richer user environment than the traditional static web pages can be provided. We will show examples of how these techniques can be used in bringing flood data to the stakeholders.

Keywords: Flood risk mapping, geo-webservices, animated mapping

1 Introduction

As a result of very high standards of flood protection, the Netherlands has a very low *probability* of flooding. Nevertheless, the *risk* of flooding (probability $\times$ consequence) can not be neglected, as this risk is larger than the sum of all other external risk sources (e.g., nuclear power plants, airports, explosions etc.). As a consequence, the importance of making the various stake-holders aware of flood risks has been stressed in various reports (RIVM 2004, e.g.).

Flood risk maps can contribute to the understanding of flood risks and raise the awareness for the consequences of floods, especially because they convey the spatial distribution of risks. The use of maps triggers the imagination of the inhabitants of an area and provides a good overview of the situation to decision makers. There are many problems in mapping risk in general (and flood risk in particular), such as the fact that we deal with fuzzy information (although mostly modeled crisply) and the fact that the information is very time–dependent and time–related. Flood risk data is in fact *spatio–temporal* data. That means for depicting these data sets it is important that the temporal aspect of that data can be visualised as such. Animation is an effective way of doing this, as we will discuss in section 2.

The group of users of flood risk maps is very heterogeneous, ranging from professionals to the general public. And the access these users have to information tools is similarly heterogeneous, from full-blown GIS to simple web browsers. Therefore making mapping solutions that serve all stake-holders well, is challenging. We argue

that solutions to some of the problems mentioned can be found in the technology that brings us geo-webservices. We will describe this technology in section 3.

At ITC we are looking into the possibility of extending the geo-webservices with the output of vector time series data as animated, interactive vector maps made within an Open Standard (and Open Source) environment. In section 4 we will present the outcomes of our efforts to build proof-of-concept applications for this, such as *TimeMapper*, a Web Map Service that outputs animated SVG maps.

2 Animated mapping

The introduction of computers in cartographic production has greatly expanded the possibilities for mapping. Changing cartography has brought new kinds of maps: We now have not only static maps, but also *interactive maps*, that can serve as a menu to the underlying data or link us to other data sources, allows to zoom, pan and even change the map design. We can go from the more traditional symbolised and generalised depictions of the world towards more realistic looking *virtual worlds* or *virtual globes*. We can combine maps with other graphics, sound and moving images and thus create *Multimedia*.

Maybe the most revolutionary new possibility is that we can now depict movement and change by using *animated or dynamic maps*. These allow us to map the time aspect of our data to the display time of our map: By showing a quick succession of slightly different images — usually called frames — a viewer sees an animation of real time. Like the spatial dimension, this time dimension is scaled. There are examples of real-time animations, where any change in reality is reflected 1:1 in the map, and we also can find examples of maps where 200 million years of continental drift are depicted in 20 seconds.

It is not strictly necessary to use dynamic visual stimuli, such as cartographic animations, in order to depict dynamic phenomena. Cartographers have been successfully mapping moving things without animations for centuries. Although the passage of time can not be visualised directly on the two dimensions of a paper map, it can be simulated quite effectively, using either single maps or map series. On single maps we can indicate movement by drawing arrows, or suggest it by drawing lines which indicate the location of phenomena at certain times. By using a series of maps we can give the viewer an impression of the passing of time or some other change from map to map. With these time–series maps it's up to the user to interpolate between the individual maps, thus giving the impression of a smooth development in time.

So it is possible to depict dynamic phenomena using static visual stimuli. Nevertheless, it appears most logical to use dynamic visual stimuli to do so. As early as 1959, Thrower pointed out that cartographers should adapt to the growing possibilities of audio-visual communication. Now that cheap and powerful computers are bringing multi-media to the homes, cartographers are almost forced into using the possibilities to "bring life" to their maps. Although people often think of animation as synonymous with motion, it covers all the changes that have a visual effect. It thus includes the time-varying position (motion dynamics), shape, colour, transparency, structure and texture of an object (update dynamics) and changes in lightning, camera position, orientation and focus and even changes in rendering techniques (Foley & al. 1990). The use of animation for maps, or cartographic animations has been the focus for research for a number of years now. From this research, we know that several types of cartographic animations can be distinguished.

Cartographic theorists have been researching the representation variables in animations. Bertin (1967) did not have much confidence in the usefulness of dynamic maps. He stated that the motion would dominate the graphic variables he distinguished (size, value, grain, colour hue, orientation and shape), thus disturbing the

effectiveness of the cartographic message. But then again, Bertin didn't like colour much either...Contrary to Bertin's opinion, recent research has shown that visual variables can indeed be used on the individual frames of an animation in such a way that these images effectively communicate the cartographic message to the user, while the movement of the animation gives the message an extra dimension and "new energy". Furthermore, the findings of Koussoulakou & Kraak (1992) show that using animated maps helps users grasp the contents of a message in a more effective manner compared to using traditional static maps or map series. It has become clear that the traditional visual variables do not suffice in describing the added means of expression we have in cartographic animations. To this end new visual variables have been introduced by DiBiase & al. (1992) and MacEachren (1994). These are called the *dynamic visual variables*, and they are: *order, moment, duration, frequency, rate of change* and *synchronisation*. It should be noted that Blok (2005) has concluded that the last two are actually not visual variables on their own, but *effects* of combining one or more of the others. The figures 1–6 (from Köbben & Yaman 1995) show examples of the application of these variables.

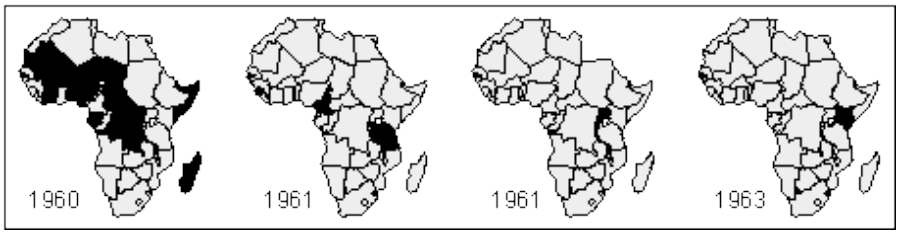

Fig. 1. Order — Animation actually is the presentation of individual frames in a given order. Chronologically showing temporal data is probably the most used form of cartographic animation. This example uses order to depict the dates of independence of African states.

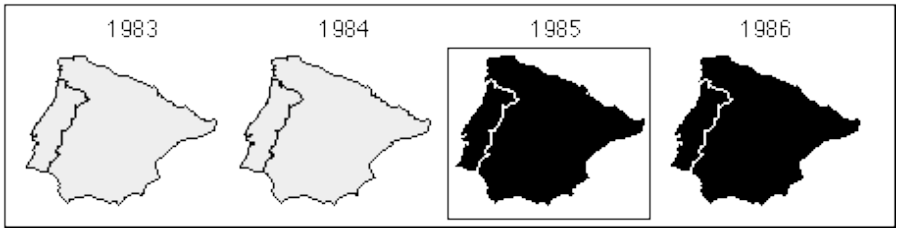

Fig. 2. Moment — The moment that an element in the map changes during a cartographic animation. An example could depict the year that Spain and Portugal joined the European Community.

One has to realise that dynamic visual variables can only be viewed in *display time* (so when the animation is "playing"), and that at least one static visual variable is required to be able to see a dynamic variable. The reason for using dynamic visual variables is that research has shown (Koussoulakou & Kraak 1992, Kraak et al. 1997, Blok 2005, among others) that there are many questions that are difficult to answer with static maps, whereas animations can provide answers to such user questions as:

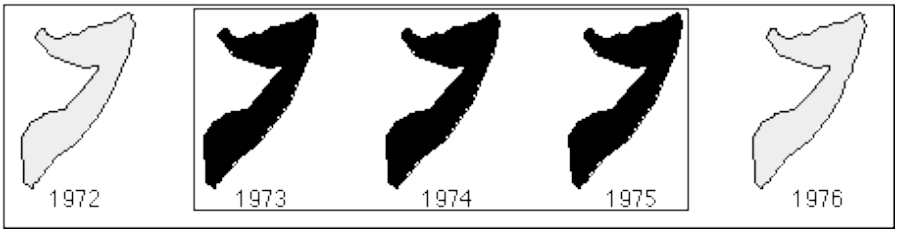

Fig. 3. Duration — Indicates the duration in display-time an element is visible during an animation. If country A experienced twice as many consecutive years of drought, compared to country B, it would be highlighted twice as long during the animation.

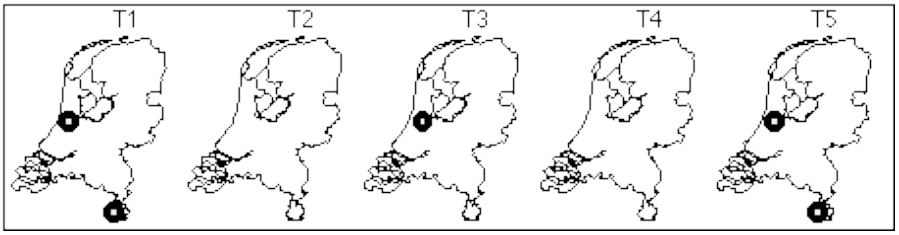

Fig. 4. Frequency — The dynamic visual variable frequency uses the rate of occurrence of graphical elements. Below, the higher "blinking" frequency of The Netherlands' main airport Schiphol indicates its bigger importance compared to Beek airport (in the southern province of Limburg).

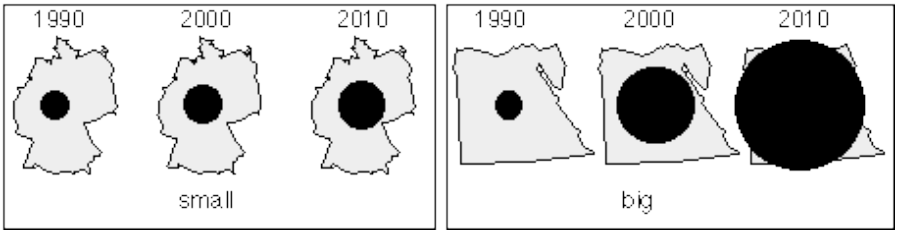

Fig. 5. Rate of change — This can be described as M/D, where M is the change magnitude and D is the duration of each scene. With change magnitude (M) the amount of change of the position and/or attribute value of an object between to subsequent scenes is described. The value of M depends on how dynamic the phenomenon is and on the pace of the animation (i.e., how many frames per minute). If M increases while D remains the same, the animation will become more abrupt and "jerky". If D increases while M remains constant, the apparent change to the viewer decreases. These are powerful tools in manipulating the viewers' perception and should be used with some care.

- when does a predicted flood reach a town?
- in what order will towns be flooded?
- how long does it take for the flooding to subside again?
- how fast does the water move?
- how often does flooding occur?
- etcetera…

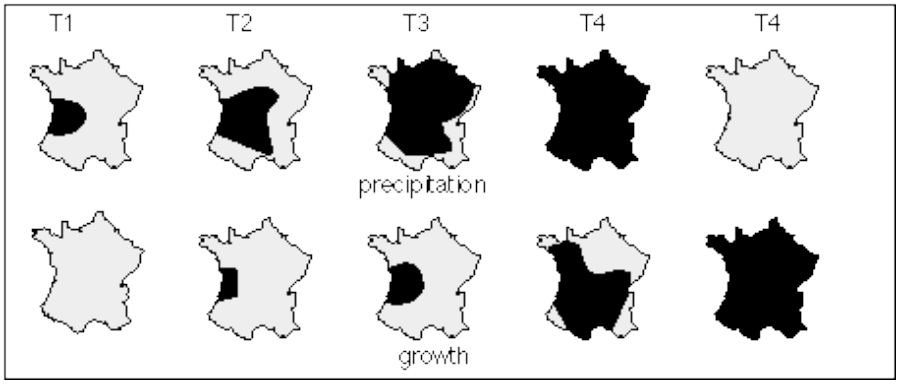

Fig. 6. Synchronisation — With synchronisation two (or more) phenomena are related to each other by showing their development synchronously in one animation. For instance, below the viewer should perceive the effects the rainfall has some time later on the growth of vegetation. This only works for fairly straightforward temporal relationships.

Answers to such question would be excellently suited for the heterogeneous user groups of flood risk maps. Cartographic theory therefore seems to offer one side of a possible solution to the problems mentioned in the introduction. But we need a way of automatically generating these animated maps from existing flood risk models and data sets and also a way to deliver the generated animations to the actual users. The geo-webservices discussed in the next section might be a way forward towards solving that problem.

3 Interoperable geo-webservices

3.1 From monolithic to distributed GIS architectures

Interoperable geo-webservices are the latest in a long line of software for handling geographic, or spatial, information. In 1962 Roger Tomlinson built the Canada Geographic Information System to determine the land capability for rural Canada by mapping information about soils, agriculture, land use, etc. He coined the term GIS for software that is used to gather, visualise and analyse geo-information (Tomlinson 1967). His GIS software was running on a large mainframe computer and its architecture was what we call *monolithic*, with the presentation logic, application logic, and data management layers combined in one software tier. This might still be the typical design for simple desktop applications, but nowadays larger GISs, like many other software systems, have their logical layers separated. This separation improves the interoperability between (parts of) the information systems: One or more database layers take care of the data storage and retrieval, application layers are used to analyse the information, a mapping engine turns the information into maps, and separate client software gives the actual users access to all of this. The main advantage of such a set-up is the flexibility: the different parts can be distributed over various computers and thus can easily be scaled, that is adopted to changing conditions, such as an increasing number of users. Because the client software is separated from the application logic, the same back-end can serve different client platforms (e.g. PCs, PDAs and mobile phones). Equally the data being used in the analysis part can come from various different information sources, from various providers, even from different places on the globe.

3.2 Webservices

Probably the best-known example of such a distributed system approach is the World Wide Web. The WWW consist of many distributed servers that are responding to individual requests of a sheer endless amount of distributed clients. This kind of system is only possible when the different components are guaranteed to work together, because they are *interoperable*. Interoperability in computer systems, as shown in Figure 7 means in the first place that the systems are able to transfer *data* seamlessly.

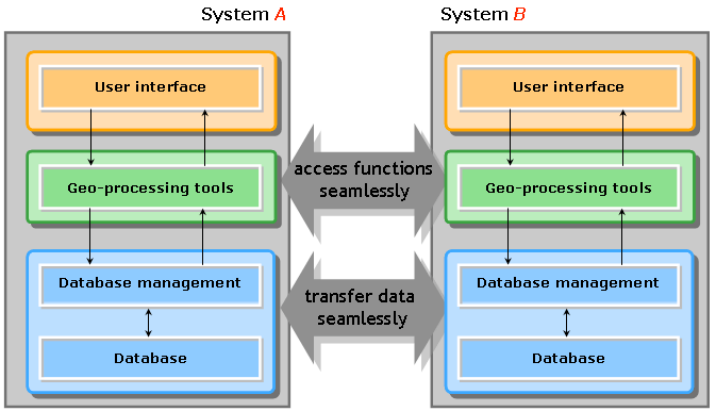

Fig. 7. Interoperable Geographic Information Systems.

This can be achieved by having the data encoded in a standardised, platform and application independent manner. Nowadays, the popular encoding scheme used for that purpose is the eXtensible Markup Language (XML).

Interoperability also means that two information systems should be able to access distributed *functionality* in a seamless manner. This would for example mean that one Geographic Information System could use the geo-processing tools of another GIS. For that to work regardless of operating system, computer platform or software used, we have to specify and set up an infrastructure of interoperable software *services*. Service-oriented software differs from traditional software in that it fully encapsulates its own functionalities and makes it accessible only via well specified and standardised *interfaces*. These interfaces publicise the methods that a software component implements, and its calling conventions. In other words, you do not know, nor have to know, how the service actually works, only what input it can receive and what output you can expect back. There are many ways of setting up such *Service Oriented Architectures* (or SOA's), but by far the most used distribution platform is the WWW and the SOA's implemented on that are logically called *webservices*.

Web Services are components that can be described, published, located and invoked over the Internet. They are defined as *loosely coupled components that communicate via XML-based interfaces* (Schmelzer et al. 2000). If webservices have *spatial functionality*, for example if they use geographic data, can output maps or find routes, we call them *geo-webservices*.

3.3 Geo-webservices

There are many such geo-webservices available, the best-know proponent of it probably being Google Maps. This, and other examples such as Yahoo Maps, Microsoft Virtual Earth or MultiMap, can be used by anybody, as their interfaces are publicly

available, but they are still *proprietary* in the sense that they are defined, developed and owned by commercial companies. Alternatively, *Open Standards* are created in an open, participatory process, where everyone interested can influence the standard. The resulting specifications are non-proprietary, that is, owned in common. That means free rights of distribution (no royalty or other fee) and a free, public, and open access to interface specifications that are also technology-neutral. There is a set of such well-defined Open Standards for geo-webservices: the *Open Web Services* (OWS) of the Open Geospatial Consortium (OGC).

The OGC was founded in 1994 as a not-for-profit, international voluntary consensus standards organisation that develops Open Standards for geospatial and location based services. Their core mission is to deliver interface specifications that are openly available for global use (OGC URL). The Open Web Services specifications are the basis of many high-profile projects (e.g., the European Community INSPIRE initiative).

There are OWS specifications for most parts of the spatial data storage, analysis and delivery process:

- for geographic data encoding: the Geographic Markup Language (GML);
- for spatial data delivery: the *Web Coverage Service* (WCS) and *Web Feature Service* (WFS), for querying and retrieving raster and vector data respectively;
- for processing of spatial data: the *Web Processing Service* (WPS). Note that this specification does not standardise the analysis or processing methods themselves, but rather defines a standardised interface that lets you publish geospatial processes, and lets client software find those processes and employ them.
- for data visualisation in the form of maps there is the *Web Map Service* (WMS). This is by far the most mature and widest adopted OWS specification. There are numerous open source as well as commercial solutions offering WMS functionality. Related to WMS are the *Styled Layer Descriptor* (SLD) specification, for map styling, and the *Web Map Context Documents* (WMCD) specification, for map set-up and layout;
- for describing and finding spatial data, there is a set of metadata specifications in the *Catalog Service Web* (CSW).

We will use WMS as an example to show the working of OGC specs in a bit more detail, see the OGC website (OGC URL) for more details on the other specifications. The WMS specification defines several interfaces, the most important are *GetCapabilities* and *GetMap*. The first is used by client software to ask for the capabilities of the service: what layers are available, what projection systems can the maps be delivered in, what output formats can be requested, etcetera. Based on this, the GetMap request is issued to ask for an actual map. Because the client knows the possibilities of the service from the GetCapabilities response, it can issue its request with specific parameters asking for example for a one or more layers of information (`LAYERS=borders,forest`), in a certain part of the world (`BBOX=xmin,ymin,xmax,ymax`), using a specific projection (`SRS=EPSG:4362`), etcetera. It will also request that the output is returned in a format that the client can handle (e.g., a web browser will request `FORMAT=image/png`), and in a specific size (`WIDTH=250\&HEIGHT=300`). Thus, if you type the URL

```
http://geoserver.itc.nl/cgi-bin/mapserv.exe?map=D:/Inetpub/
    geoserver/mapserver/config.map&SERVICE=WMS&VERSION=1.1.1&
    REQUEST=GetMap&LAYERS=forest,railroad,airports&STYLES=&SRS
    =EPSG4326&BBOX=97,5,105,20&WIDTH=400&&HEIGHT=600&FORMAT=
    image/png
```

in a web browser, the WMS at the site `geoserver.itc.nl` will return the graphic shown in Figure 8. Of course these URLs are normally not typed by humans, but put together and processed by client software. If the user of a mapping client triggers a zoom command, the software will issue a new GetMap request, now with a smaller `BBOX`. This process will be repeated while the user interacts with the map.

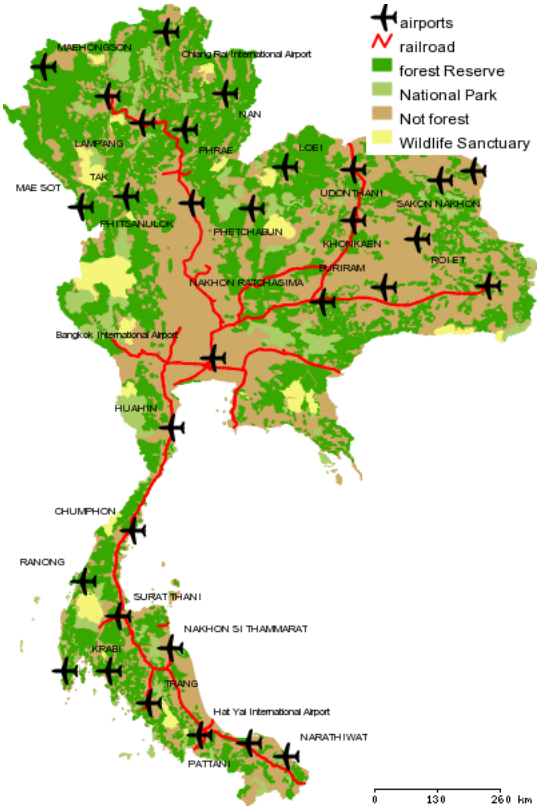

Fig. 8. Result of the WMS request in the URL on page 7

3.4 Interactive flood mapping applications using geo-webservices

At ITC, we use geo-webservices in teaching, research and consultancy. Students and staff build services and clients using a software stack that is employing so-called Free and Open Source Solutions for Geospatial data. The *Open Source Geospatial Foundation* has been created to support and build high-quality open source geospatial software, most of the software mentioned here is part of that effort and can be found at the OSGEO website (OSGEO URL).

One test we have done using this software stack is the site shown in figure 9, set up to demonstrate using standard geo-webservices to bring flood data to stakeholders. We can combine the WMS providing the flood data with any other WMS for additional layers. In our example we do this with general geodata of the Netherlands coming from OpenStreetMap (OSM URL). This data includes road data up to street level, land use data, hydrography and built-up areas. And because of the inherent interoperability of the WMS, any other geo-webservice could be combined with the flood data, including proprietary ones such as Google Maps. We built a simple mapping client in a web page, that enables you to load various layers, switch them on or off, zoom and pan the map and also to click in the map to find attributes for the various data layers for that location. This solution based on existing WMS software does offer many of the things we are looking for, but it only allows for raster output formats with very limited interactivity and no animation possibilities.

Although WMS implementations generally render their maps in a raster format such as PNG, GIF or JPEG, a small proportion of them also offer vector-based maps using the word Wide Web standard vector graphics format Scalable Vector Graphics

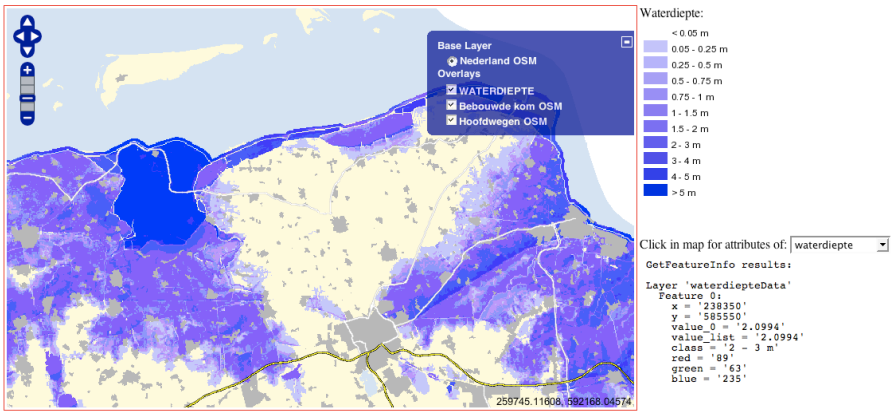

Fig. 9. Modelled water depths after dike-breach in northern Netherlands. Interactive original can be found at `http://geoserver.itc.nl/cartesius/`.

(SVG). However, these implementations all treat this output as just another graphics format, and like the raster output, the maps are basically pictures only, with no interactivity or 'intelligence'. But SVG is more than a graphics format only, it also offers interactivity, through built–in scripting and full access to its (XML) Document Object Model. It is therefore possible to build an SVG map, or rather an SVG *application*, that includes its own user interface as well as interactive data–access. This has been demonstrated by others in various examples (see e.g. Carto:Net site URL).

However, these solutions have been created on a case–to–case basis, usually as stand–alone applications, and they are not easily generated by or deployed repeatedly from a dynamic set of data. To overcome this, we developed *RIMapperWMS*, set up to make such a solution more generic, by offering a simple WMS conformant interface to the spatial data that outputs interactive SVG applications. Examples of its use and a detailed description of its set-up can be found in Köbben (2007) and on the RIMapper website (RIMapper URL). The example output in figure 10 shows a map that indicates flood risk in Ward 20 of the city of Kathmandu (Nepal). A 'fuzzy' risk symbol, growing and shrinking while moving the mouse, shows the underlying flood risk data. This gave us the additional interactivity and data–access we were looking for, but not the animations…

4 Combining geo-webservices and animated mapping

The OGC standards deal with spatio–*temporal* data, thus there are provisions in the WMS specification for handling time–series. A TIME parameter can be used in the URL request, this parameter employs a standardised *time format* (up to 14 digits specifying century, year, month, day, hour, minute, and seconds) and a *period*, used to indicate the time resolution. For example, `P1Y` means a period of 1 year. Thus, the URL parameter `TIME=2010-07-01T0:00/2010-07-01T24:00/P1H` requests hourly data for the 7th of July 2010. TIME can then be used in conjunction with the FORMAT parameter to request time series output.

Currently WMS implementations exist that produce *movie–like* animations of *raster* based (earth observation) data. E.g., a request for daily ozone maps for a specified period that results in an MPEG movie. These movies are limited in their interactivity to basic VCR–type interactions, such as play, pause, fast forward, etcetera. Furthermore, the individual frames of the movies are just pictures, and have lost any connection with the underlying data or models.

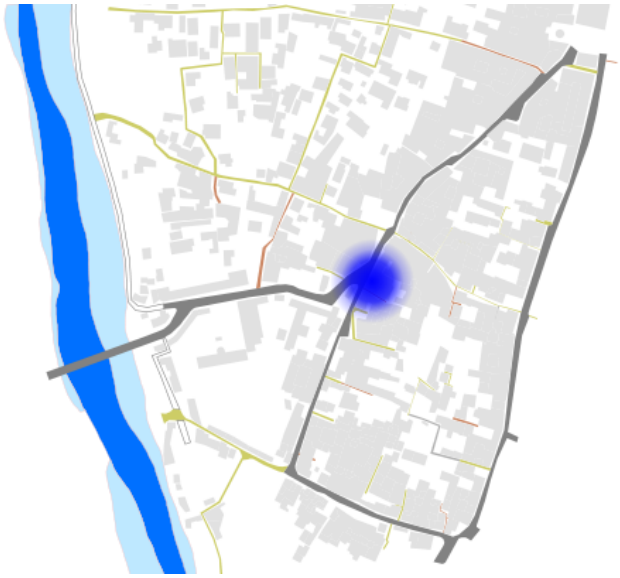

Fig. 10. Flood risk in Kathmandu Ward 20. Interactive original can be loaded from `http://kartoweb.itc.nl/RIMapper/SVGopen2003/FloodsMap.svg`

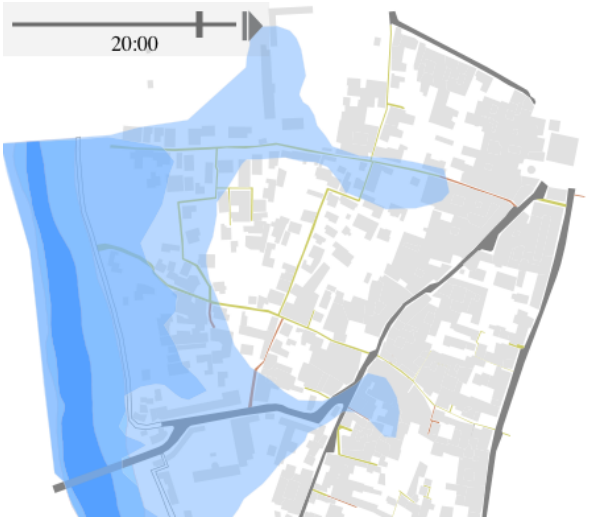

Fig. 11. Animation of modeled flooding times in Kathmandu Ward 20. Interactive original can be loaded from `http://kartoweb.itc.nl/RIMapper/TEST/FloodsAnimated.svg`

We have recently been looking into the possibilities of extending WMS services with output of *interactive vector* animations, based on the RIMapperWMS framework described above. The SVG format offers two basic types of animation: Programmed, or *procedural* animations can be made using JavaScript manipulation of the SVG graphics. An example of this can be found in the experiment shown in figure 11. Here the same flood risk data of Kathmandu we used earlier is now shown as an animation of predicted flooding polygons. These are shown in a timed sequence, either by au-

tonomous playback or by interactively moving the time slider. Mouse–overs provide tool-tips of the underlying attributes, in this case the predicted period of flooding.

But the SVG standard also includes the Synchronised Multimedia Integration Language (SMIL), an XML-based language for animation of a wide range of various graphic elements. SMIL's *declarative* syntax is compact and powerful. The following code shows a simple animate element: `<animate attributeName="width" from="100px" to="800px" begin="click" dur="7s" />` This animate element could be embedded within an SVG shape such as a rectangle. When the user clicks the shape, a smooth animation will change the width of the shape from 100 pixels to 800 pixels over a duration of 7 seconds. These SMIL animations are much more sophisticated than the procedural JavaScript ones, for example because it offers built–in interpolation mechanisms, both spatial and temporal. It allows us to build full–fledged animation environments, with interactive control of the display–time, as a scaleable depiction of real–world time.

In recent months, we have been looking into extending the RIMapperWMS framework to enable generation of these SMIL–based animation applications in an automated, data–driven manner from WMS time–series output. A first proof–of–concept application has been constructed as part of the practical work of an ITC MSC-thesis (Becker 2009). An example of output of the extended framework, which we call *TimeMapper*, can be found in figure 12.

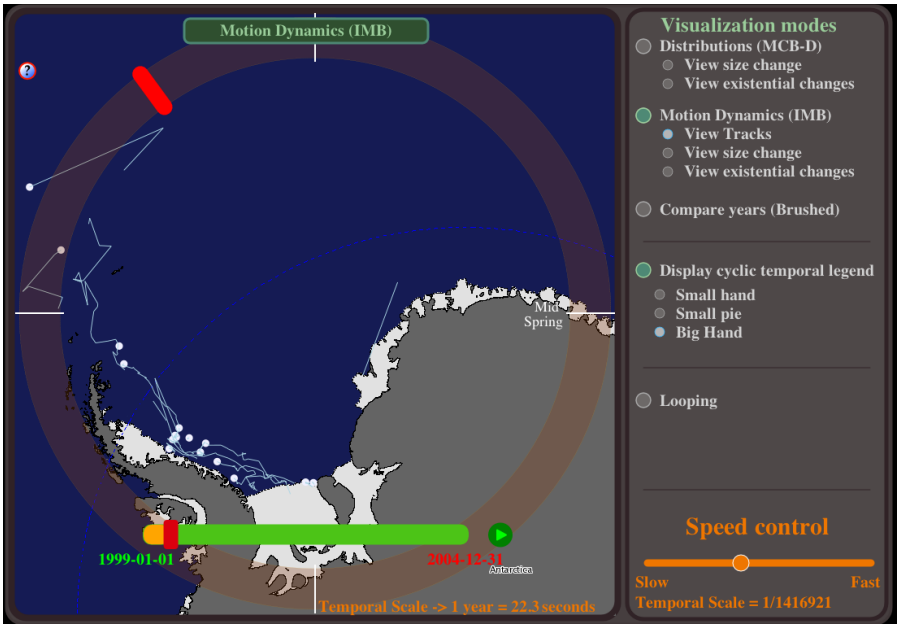

Fig. 12. Prototype of interactive SMIL animation of 19 icebergs in the Weddel Sea (Antarctica) from January 1, 1999 to December 31, 2004. The animation application offers various modes of animation. The currently selected one is 'Motion Dynamics', which shows not only measured positions (actually in the data), but also interpolates intermediate positions and shows the interpolated tracks. Various temporal legends are available, of which the 'Big Hand' is shown: A transparent clock–like ring, whose hand runs around once every real–time full year. Interactive original can be loaded from `http://geoserver.itc.nl/TimeMapper`

.

5 Conclusion

Apart from showing the powerful visualisation possibilities of animation in general, and of SMIL–based SVG animation specifically, the TimeMapper prototype also demonstrates the overall principle of *interoperability* our approach offers: The animated iceberg symbols and tracks are generated by our TimeMapper application, whereas the Antarctic base map is made up of several layers of information coming 'live' from external sources, in this case the WMS services of the Antarctic Cryosphere Access Portal of the US National Snow and Ice Data Center.

The animations are generated 'on–the–fly' from the underlying spatial database and therefore will always reflect the most up-to-date version. Furthermore, this allows the user to get access to this data through the application, e.g., to show attributes of objects clicked, or to filter specific sub–sets to be visualised.

Next in our plans is the application of the TimeMapper system to map the extensive flood–modelling of the "Land van Maas en Waal", a Dutch river region prone to dike-failures. At ITC we have a large data set of these models, from research by Alkema (2007). Previously, animations of the data have been constructed as stand–alone applications based on Adobe's proprietary Flash software, with no link to the actual data sets and models. We intent to test if TimeMapper can be used to render more interactive, interoperable and flexible animations of these flood–risk models.

Acknowledgements

Section 3 of this paper is based on the unpublished lecture note "A short introduction to geo-webservices", used in teaching at the International Institute for Geo-Information Science and Earth Observation (ITC). This in turn is based on the teaching materials (slides, exercises) for various modules on WebGIS, Webmapping, Spatial Data Infrastructures and the like, developed partly by the author of this paper, but in large parts also by his colleagues Rob Lemmens, Andreas Wytzisk and Javier Morales, for which they are gratefully acknowledged.

References

Alkema, D. (2007), Simulating floods - On the application of a 2D-hydraulic model for flood hazard and risk assessment, ITC dissertation no. 147, International Institute for Geo-Information Science and Earth Observation.

Becker, T. (2009), Visualizing time series data using web map service time dimension and SVG interactive animation, Msc thesis, International Institute for Geo-Information Science and Earth Observation.

Bertin, J. (1967), *Sémiologie Graphique*, Mouton, Paris/Den Haag.

Blok, C. A. (2005), Dynamic visualization variables in animation to support monitoring of spatial phenomena, PhD thesis, ITC.

Carto:Net site (URL), 'SVG navigation tools tutorial', `http://www.carto.net/papers/svg/samples/`, last accessed Dec 2008.

DiBiase, D. & al. (1992), 'Animation and the role of map design in scientific visualisation', *Cartography and GIS* **19**(4), 201–214.

Foley, J. & al. (1990), *Computer Graphics: Principles and practice*, Addison-Wesley, Reading.

Köbben, B. (2007), RIMapperWMS: a Web Map Service providing SVG maps with a built-in client, *in* S. Fabrikant & M. Wachowicz, eds, 'The European Information Society - Leading the way with Geo-information', Lecture Notes in Geoinformation and Cartography, Springer-Verlag, Berlin, etc., pp. 217–230.

Köbben, B. & Yaman, M. (1995), Evaluating dynamic visual variables, *in* F. Ormeling, B. Köbben & R. P. Gomez, eds, 'Seminar on Teaching Animated Cartography', International Cartographic Association, Madrid/Utrecht, pp. 45–53.

Koussoulakou, A. & Kraak, M. (1992), 'Spatio-temporal maps and cartographic communication', *The Cartographic Journal* **29**(2), 101–108.

Kraak, M. J., Edsall, R. & MacEachren, A. M. (1997), 'Cartographic animation and legends for temporal maps : exporation and or interaction', *In: ICC 1997 : Proceedings of the 18th ICA International cartographic conference : 23-27 June 1997, Stockholm, Sweden. International Cartographic Association (ICA), 1997. Vol. 1. 8 p.* .

MacEachren, A. (1994), *How maps work: Issues in representation & design*, Guildford Press, New York.

OGC (URL), 'Open Geospatial Consortium Home Page', `http://www.opengeospatial.org/`, last accessed Jan 2009.

OSGEO (URL), 'Website of OSGEO foundation', `http://www.osgeo.org/`, last accessed Jan 2009.

OSM (URL), 'Open Street Map website', `http://www.openstreetmap.org/`, last accessed Jan 2009.

RIMapper (URL), 'RIMapper project pages', `http://kartoweb.itc.nl/rimapper/`, last accessed Feb 2009.

RIVM (2004), Risico's in bedijkte termen, Technical Report 500799002 (ISBN 90-6960-110-9), Rijksinstituut voor Volksgezondheid en Milieu, Bilthoven.

Schmelzer, R., Vandersypen, T., Bloomberg, J., Siddalingaiah, M., Hunting, S., Qualls, M., Darby, C., Houlding, D. & Kennedy, D. (2000), *XML and Web Services Unleashed*, 1st edn, SAMS publishing.

Thrower, N. (1959), 'Animated cartography', *The professional geographer* **11**(6), 9–12.

Tomlinson, R. (1967), *An introduction to the Geo-information system of the Canada Land Inventory*, Dept. of Forestry and Rural Development.

****4008 words****