

Chapter 14

Webservices for Animated Mapping: The TimeMapper Prototype

Barend Köbben, Timothée Becker, and Connie Blok

Abstract Within a larger aim of improving automated vector animated mapping, the main objective of this research was to look into the possibility of combining two technologies: distributed webservices and animated, interactive vector maps. TimeMapper was developed as a prototype for an OGC-compliant Web Map Service implementation that serializes spatio-temporal data from a database backend as Scalable Vector Graphics. The SVG is used in a web browser to show animated maps with a built-in advanced user-interface. This interface allows the user to interact with both the spatial and the temporal dimensions of the data. The potential and limitations of the TimeMapper prototype were explored using Antarctic iceberg movement data. The prototype can be explored on the TimeMapper website (<http://geoserver.itc.nl/timemapper/>).

14.1 Introduction

The motivation for building the TimeMapper prototype was the improvement of automated animated vector mapping. We are interested in animation, because interactive animated mapping has been pointed out, by Andrienko et al., (2003) among others, as the only technique to be generically applicable to visually analyze the dynamic nature of real world phenomena.

We want to facilitate the production of animated maps from spatio-temporal data to a format suitable for Internet dissemination, *automatically* and *directly*. To achieve that, we looked specifically into the possibilities of the loose coupling of distributed webservices with animated, interactive vector maps.

B. Köbben (✉)

Faculty of Geo-Information Science and Earth Observation, ITC – University of Twente,
Enschede, The Netherlands
e-mail: kobben@itc.nl

14.1.1 Animated Mapping in a Webservices Environment

The principle of disseminating maps in a webservices environment is depicted in Fig. 14.1. This general set-up is being used in many of today's webmapping efforts, with considerable variation in the choice of technology for the mapping service (depicted as A) and the subsequent map formats (B). This choice to a large extent defines the possibilities of the system as a whole to achieve what we called previously the *automatic* and *direct* production of animated maps.

We consider "automatic" to mean that the animated maps will be generated from the spatio-temporal data by the system "working by itself with little or no direct human control" (which is how "automatic" is defined in Fowler et al., (1976)). It is important to note that this automation will not include the cartographic decisions as to what type of map to use for different data-types and data-instances. As in most current systems, this level of automation in cartography will not be achieved. The link between data- and visualisation-type has to be made by a human (the cartographer in Fig. 14.1), setting up the appropriate configuration parameters beforehand.

By "direct" we mean that the maps are generated on-the-fly from the data, without conversion or pre-processing needed for the purpose of visualisation only. This is important because we want our system to fit in an interoperable Spatial Data Infrastructure, where it would be an SDI node, as defined in de By et al., (2009): "a single system node in an SDI network, which archetypically has data, catalog or portrayal services." In such a context, our system would be just another service, that should be able to consume data from any other SDI node, and whenever that data changes, the maps generated from that data would change. To achieve this, the visualization functionality should be loosely coupled to the other parts of the system. The Open Geospatial Consortium's (OGC) Web Map Service (WMS) and related specifications (see OGC 2002b, 2006) are especially useful for this. These webservices are designed to take their input from a variety of distributed sources and generate output meant for Internet dissemination.

Current WMS implementations mostly generate so-called "picture" output, such as the raster formats PNG, GIF and JPEG. These picture formats are fine for

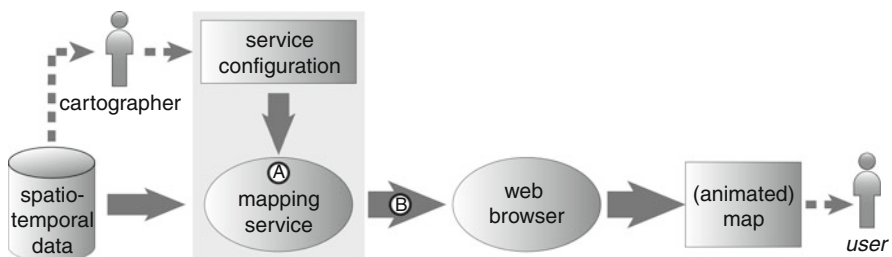


Fig. 14.1 General principle of dissemination of maps in a webservices environment. Note the important influence, discussed in Sect. 14.1.1, of the choice of technology and the subsequent formats in A and B. In this research A is the TimeMapper WMS prototype, and B is Scalable Vector Graphics XML, including SMIL animation and interactivity

delivering static map images over the web. For data that has a temporal component, a series of such static images, packed into some sort of animation format (e.g., an animated GIF or a QuickTime movie) can be generated; this is supported by several existing implementations of WMS. These raster animations, comparable to movies, can be useful for certain types of temporal datasets, especially satellite image series.

However, this type of animation lacks the possibility to express many of the aspects of temporal changes that are connected to specific *objects*, not fields. In geographic theories on change, different types of change are usually recognized. Based on the work of Blok (2005), Andrienko and Andrienko (2006) describe three categories of change: existential change (e.g., appearance/disappearance), change of spatial properties (e.g., location, shape) and change in attribute (e.g., ownership change). These changes occur over time, to selected geographic objects, that are in many cases modeled in the data as polygons, lines and points.

To be able to express these changes on the object level, we are interested in *vector* animations. The WMS specification also allows for “graphic element” output, i.e., using vector formats with added possibilities, such as scripting and animation. The added benefit of this is that such formats allow the inclusion of attributes within the objects, providing the client-side mapping application access to data attributes without the need for a round-trip to the server.

At the time of our research, two graphic element formats were widely used to produce vector animations for the Internet: Scalable Vector Graphics (SVG) and Adobe Flash. SVG is our format of choice: firstly because it is an XML-based Open Standard (W3C 2010), whereas Flash is a binary format and a proprietary technology. Secondly, because we have good experiences with a previously built SVG-based WMS for static maps. And thirdly, because the SVG specification allows the use of the Synchronized Multimedia Integration Language (SMIL), a declarative XML-based language well-suited for building data-driven animations (W3C 2008).

14.1.2 Objective and Set-Up

The **objective** of our research was therefore *to design an animated mapping system which can generate SVG SMIL animations, automatically and directly, from spatio-temporal data*. The scope of vector animated mapping is broad and we therefore limited our initial goal by developing a prototype system for one specific type of real-world phenomenon only: the movement of objects in geographic space.

To achieve our objective, we extended our existing SVG-based Web Mapping System for static maps with the animated mapping functionality. We briefly describe this existing system, called RIMapperWMS, as well as other related work, in Sect. 14.2. We then explain the extensions necessary to create animated maps and their set-up and introduce our prototype, called TimeMapper, in more detail in Sect. 14.3. We evaluate the prototype in Sect. 14.4 and, based on that, draw some conclusions and give an outlook on further work in Sect. 14.5.

14.2 RIMapperWMS and Other Related Work

Our starting point for the TimeMapper prototype was RIMapperWMS, which in turn was a further development of RIMapper. This was a non-standardised webservice, that started out as the visualisation part of an urban risk management system, hence the name, an acronym for Risk Indicator Mapper. RIMapperWMS, described in more detail in Köbben (2007), was an effort to make the software compliant with OGC's Web Map Service (WMS) specification. Like most of the software we use, it fits within the so-called SDI^{light} approach, that we have been using for several years in our teaching and research at ITC. It has been discussed in detail elsewhere, most recently in Köbben et al., (2010). In short, it is a down-to-earth approach towards Spatial Data Infrastructures, using Open Standards whenever available, and open source solutions where possible. This approach provides researchers, students and partners with a platform for relatively simple, low-cost, yet powerful ways of sharing data amongst various stakeholders.

The main components of the platform are firstly a spatial database back-end (we most often use the object-relational DBMS PostgreSQL and its spatial extension PostGIS). Secondly, there are middleware web applications that interface with the database back-end and with each other, and fulfill tasks such as delivering maps, data and processing services. We use existing open source applications (mostly MapServer and GeoServer), but also develop our own components. The RIMapperWMS and TimeMapper discussed here are examples of those. And, finally, there are open source desktop clients (such as QuantumGIS and ILWIS), as well as browser-based clients (such as OpenLayers and the SVG GUI part of TimeMapper), that enable access to the maps and data.

There are several established WMS implementations that serve SVG. However, based on our own experiences and the capabilities as described on the OGC implementation webpage (OGC 2002a), they treat the map output as just another graphics-only format. So, just like the raster format output in PNG and JPEG and such, the maps are basically pictures only, with no interactivity or "intelligence". To have the map behave like a *mapping application*, it needs to be wrapped in a Graphical User Interface (GUI) that takes care of zooming, panning and other interactions. There are many such GUI's available, as dedicated server-side applications, as part of general GIS clients (such as QuantumGIS), or as client-side API's for browser environments (such as the popular OpenLayers Javascript API). But, because 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 deliver straight from the WMS an *SVG application*, that includes its own GUI. This is the strategy we use in RIMapperWMS, and its output can therefore be used without a dedicated mapping client, any SVG-enabled client is sufficient. At present, this includes most major web browsers: Firefox, Opera, and WebKit based browsers (Chrome, Safari, etc.). Until recently one needed to use a plugin for Internet Explorer but the latest IE 9 release now supports SVG.

Since introduced in 2007 (Köbben 2007), the RIMapperWMS software has been further developed to version 1.0, and is now a fully functional and compatible version 1.1.1 Basic Web Map Service. The GUI has been improved by including a layer switcher and an info tool, and is now draggable. A transcoder has been added for the ability to output PNG and JPEG, for increased interoperability and cascading WMS ability. RIMapperWMS is published under an open source license and the components, their source, as well as some documentation can be found on the RIMapperWMS website (ITC 2010a).

Other researchers have also looked into the added possibilities of SVG in a WMS environment. In the work presented by Ostländer et al., (2005), the SVG is only used to build an interactive mapping client, the maps are traditional WMS raster output wrapped in the SVG client-side. Behr et al., (2006) present a WMS with dedicated SVG-based clients (both for browsers and for mobile clients), but these client GUIs are not included in the SVG output of the WMS. The system they describe actually stores its data in SVG format. This does diminish the possibility for interoperability and loose-coupling of the WMS in a SDI environment as described in Sect. 14.1.1. Their work has been further developed into the SUAS Mapserver (see EasyWMS 2011), a project with a much broader focus than WMS only, but which does not include animated mapping.

Professor Yan Li and a group of researchers at the Spatial Information Research Center of the South China Normal University are working on a complete GIS solution based fully on XML–GML–SVG (Li et al., 2009). It is a wide-ranging project involving much more than just map services, but like Behr’s prototype, it is not intended to function loosely-coupled in an SDI environment and does not include animated mapping.

Brinkhoff (2007) describes an interesting use of SVG in combination with GML and SLD for what he calls “WMS mashup services”. Brinkhoff’s architecture is actually focussed on a *tight* coupling to allow users to add their own geospatial objects to WMS *images*. Although he mentions the use of SMIL for image maps, it is not used for animating the WMS layers.

We therefore feel that the prototype described in the next section is the first effort to build an OGC-compliant Web Map Service implementation that serializes spatio-temporal data from a database backend as Scalable Vector Graphics with SMIL animation.

14.3 Animated SVG Maps from a WMS: The TimeMapper Prototype

The TimeMapper prototype was designed to add animated SVG and associated GUI extensions to the existing RIMapperWMS architecture. To focus on the animated movement of objects in geographic space, we chose the visualisation of the dynamics underlying the calving and spread of icebergs. The data used is the U.S. National Ice Center (NIC) Antarctic icebergs dataset.

14.3.1 Test Case Data: Antarctic Icebergs

This data consists of movement data, gathered since 1978, of 301 icebergs (with currently 15,737 records) that can be downloaded from the NIC website (National Ice Center 2010). Besides the iceberg position in latitude and longitude, and the time of recording, the dataset contains basic descriptive information (name, size, etc.). The temporal resolution is quite irregular, ranging from 1 day between consecutive positions, to up to 50 days. The iceberg dataset has been used by different researchers with various objectives. Many have described the importance of visualising the dynamics of movement as well as calvings or splits, appearances and disappearances, of the icebergs. For example, Schmittner et al., (2002) used these as climate change indicators, and Benn et al., (2007) looked at the relationship with external factors, e.g., winds and ocean currents. To be useful for visualising the properties and relations of the data, the TimeMapper prototype would need to show animated point symbols (depicting icebergs), together with relevant topographic data, to provide an appropriate spatial reference, as well as other relevant data, such as ocean currents.

14.3.2 Extending RIMapperWMS

To extend the RIMapperWMS system to achieve the mapping of temporal data as SMIL SVG animations, we addressed the following points:

- Firstly, we designed SVG SMIL animation types we could use to represent the real-world dynamics;
- Secondly, we needed to develop an algorithm for converting the temporal component of the data, compliant with relevant standards, to the temporal designations in the SMIL format within SVG;
- Thirdly, we determined, based on animated mapping literature and experiments, which user-interface elements and which interactivity we would offer to the user, and how those could be implemented in an SVG environment;
- Finally, an appropriate architecture for the overall system was designed.

14.3.2.1 Designing the SVG SMIL Animations

In our present test case, we focus on *point* objects, i.e., the icebergs, and their changes in *position* over time. To show the movement of the objects, two main types of animations can be used, *discrete* animations and *interpolated* animations. Such animations are easy to declare using SVG SMIL. The following code snippet shows an example of how SMIL declaration works, in this case for an animation for the step-wise movement of an iceberg symbol:

```

<circle id="IB_A22B" cx="0" cy="600" r="25" >
  <animate id="XanimIB_A22B_0" attributeName="cx"
    repeatCount="none" fill="freeze" begin="0s"
    from="0" to="200" dur="2s" calcmode="discrete" />
  <animate id="YanimIB_A22B_0" attributeName="cy"
    repeatCount="none" fill="freeze"
    from="600" to="550" dur="2s" calcmode="discrete" />
</circle>

```

SMIL animations are defined by including them as children of SVG elements, in this case of the circle element depicting iceberg IB_A22B. The first animate element will lead to movement along the x-axis, the second along the y-axis. The begin attribute tells the animations when to start, and the dur attribute sets its duration and therefore its apparent speed. The calcMode attribute is set to “discrete”, which means we will see a stepwise animation. In brief, the effect of these two animations is that the object is going to jump from 0 to 200 on the x-axis and at the same time from 600 to 550 on the y-axis after a count of 2 s. For a linearly interpolated animation, the calcMode attribute would have to be changed to “linear”, and then the movement would be gradual over the course of the 2 s.

In our moving-object prototype, the interpolated animation type is offered to the users for them to be able to appreciate the movement of the objects taken individually or in small groups. In other words, we are interested in the dynamics of the *trajectories*. To help visualise this, we added animated tracks. SVG SMIL doesn’t have a standard way of animating movement with line segments. However, members of the SVG community have developed a workaround by animating the stroke-dasharray attribute of lines, resulting in lines that are “growing” over time. The result can be seen in Fig. 14.2 as the thin lines trailing the circles depicting the icebergs. As can be seen in the code example above, the SMIL animations use a timing mechanism based on display time, that is defined in seconds. Therefore the real-world time in the spatio-temporal data has to be converted to appropriate values.

14.3.2.2 Converting the Temporal Dimension of the Data

For loose coupling of the WMS with data from arbitrary SDI nodes, we have to assume the temporal component of the data is made available in a standardised format. We therefore followed the relevant OGC specifications, which in turn are based on the ISO 8601 standard. This defines the encoding of time stamps and time periods in single strings, following the format ccyy-mm-ddThh:mm:ss.sssZ: Up to 14 digits specify century, year, month, day, hour, minute, seconds, and optionally a decimal point followed by zero or more digits for fractional seconds, with non-numeric characters to separate each piece. All times are expressed in Coordinated Universal Time (UTC) as indicated by the suffix Z (for “zulu”). Thus the timestamp 2009-10-04T14:50:58Z indicates October 4, 2009 at 2:50 pm and 58 s. In the



Fig. 14.2 Screenshot of the prototype of interactive animation of 19 icebergs in the Weddel Sea (Antarctica) from January 1, 1999 to December 31, 2004. An interactive version can be loaded from the TimeMapper website (<http://geoserver.itc.nl/timemapper/>)

previous section, we showed how display time is used in SMIL animation. To go from real-world time in the data to display time in the animations, the first step is to convert the ISO 8601 time stamps to a single unit time format, in our case, the number of seconds since an arbitrary fixed starting point. Next comes the scaling of time, by multiplying all time-stamps by a ratio that we call the *temporal scale*. Just as the traditional cartographic spatial scale is the ratio between map distances and real-world distances, the temporal scale in animated cartography is the ratio between display-time and real-world time (Harrower and Fabrikant 2008). Most often, as in our prototype, time will be shrunk for the viewer to visualize in a short time the events which happened during a comparatively long period.

14.3.2.3 Developing the Animated Mapping User Interface

Most theorists agree that the users of animated maps should be helped in their understanding of the information by two elements: temporal legends and ways to interact with the temporal dimension of the maps. Kraak et al., (1997), among others, state the importance of temporal legends and recognise three main types of temporal legends: digital clocks, time bars and cyclic temporal legends. All three of these legend types were developed for our prototype.

Various interactive functionalities were developed for the graphic user interface (GUI): to enable the user to choose between stepwise and interpolated animations, to add or remove interpolated tracks, and to choose which cyclic legend to view (if any). To control the temporal dimension, both a time scaling mechanism and time control tools were devised. The mechanism behind these interactive tools is a series of Javascript functions embedded in the SVG, changing the attributes of the various elements that make up the GUI and the map.

Various elements of the GUI are presented in Fig. 14.2: the *Big Hand* cyclic temporal legend has been chosen, and it is visible as a transparent circle over the whole map image with a red bar that travels the full 360° in 1 year of real-world time. By selecting *Motion Dynamics*, the user has chosen interpolated animations. The *View Tracks* option has resulted in the moving point symbols depicting the icebergs (the white circles) having their tracks traced with narrow lines. The *Speed control* on the lower-right side lets the user adjust the temporal scale, and the *time control bar* on the lower-left is for stopping, starting and brushing time.

There is limited space here to go into details on the choices for, and set-up of the various GUI elements. GUI design was based on a literature review and experiments that were part of an ITC MSc research and are described in more detail in Becker (2009).

14.3.2.4 System Architecture

To bind all elements together we used an overall system architecture that is basically an extension of RIMapperWMS (see Sect. 14.2). We therefore provide only a general overview here, as depicted in Fig. 14.3.

The core of the application tier is a set of Java Servlets that can be deployed in any J2EE compatible servlet container. The servlets do recurring tasks like extracting OGC features and attribute data from the database, translating these into fragments of SVG, SMIL and JavaScript, collecting and structuring these fragments into valid output and delivering this output to the clients. The web interfaces of the servlets comply to OGC's Web Map Service 1.1.1 standard (OGC 2002b): Firstly, *GetCapabilities*, returning an XML description of the WMS information content and acceptable request parameters and secondly, *GetMap*, returning the map itself.

The WMS is configured using a database back-end. In our prototype, this database is a set of PostGIS tables, and it can also include the spatial and attribute data to be mapped, stored according to the OGC "Simple Features for SQL" specifications (OGC 1999). Other spatio-temporal data can be retrieved from external SDI-nodes, but this has not been implemented in the current prototype.

Based on a WMS GetMap request, the servlets generate an SVG file that defines the setup of the map and its GUI: the necessary scripts for the basic WMS GUI as well as the animation GUI are loaded, and the map layers are defined as a set of nested SVG elements.

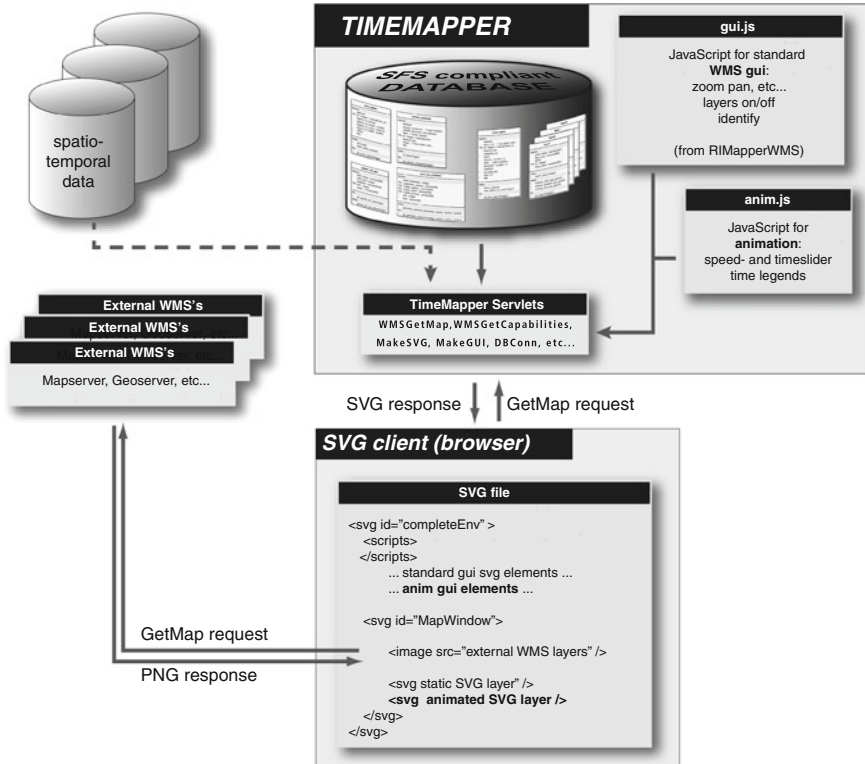


Fig. 14.3 TimeMapper system architecture overview. Note that the *dashed* link with external spatio-temporal data is not used in the current prototype

Note that the content of the map layers is not limited to the animated SVG generated by TimeMapper. External WMS layers can also be included, for instance, in the example in Fig. 14.2, the land and ice shelf are raster WMS layers from the Antarctic Cryosphere Access Portal OGC services (<http://nsidc.org/agdc/acap>). The GetMap requests for these external WMS's are encoded in the src attribute of an SVG image element, and subsequently retrieved asynchronously by the browser. And, of course, additional non-animated SVG layers can be retrieved from RIMapperWMS services.

14.4 Evaluation of the Prototype

With our TimeMapper prototype, a series of three test-cases was generated, all depicting the same 5 year period (1999–2003), with an increasing number of icebergs displayed. The *small* test case shows three icebergs, with a total of 218 separate data entries, or occurrences. The *middle* test case has 19 icebergs,

i.e., all the icebergs present in the Weddell Sea region during the 5 years, accounting for 811 occurrences. The *big* test-case shows 99 icebergs, including all icebergs in the whole Antarctic Ocean during the 5 years, totalling 4,825 occurrences.

Testing was done using the Opera web browser, as the only web browser supporting the full SVG SMIL specification at the time of testing. Other browsers are nearing full compatibility at the time of writing, Mozilla Firefox and browsers based on the Webkit engine (e.g., Safari and Chrome) support almost all of TimeMapper's functionality. Microsoft recently announced that it will implement SVG natively starting with Internet Explorer 9, although it currently looks like SMIL animation will not be supported in the first version.

While testing the prototype with the small test case, we found all the features working as planned. The speed setting system (setting the time scale) as well as the time control slider work particularly well, and are very responsive. The user can interact with the temporal settings of the animation with hardly any delay, creating a true time-brushing effect.

However, with larger test-cases, there are important responsiveness limitations, that mostly affect the time scaling mechanism and the time control slider. Changing the speed of the animations with the middle test case takes a few seconds, after which the animations themselves work fine. But, although interacting with the time slider while the animation runs still works, it is too slow to constitute a usable exploratory tool. Finally, the big test case takes a long time (over 60 s) to load. Setting the speed of the animations and interacting with the time slider hardly works at all, although once it is finally running, the animation is still reasonably smooth.

In recent meetings at the SVG Open developers conference (<http://svgopen.org>), the performance issues were discussed with SVG implementors. The lack of performance in interactivity seems to be mainly due to the fact that the browser needs to manipulate large amounts of objects in the DOM tree that represents the SVG in memory. DOM manipulation is a recognized performance problem in current browser implementations. With our present animation set-up, each segment travelled by an object leads to two animation objects (one x and one y) per segment travelled. Testing is planned using an alternative coding method for the animation, such as `keyTimes` and `KeyValues`. With these SMIL methods, the whole trajectory (i.e., the sum of the segments) travelled by an object would lead to only two animation objects and we feel this may improve the responsiveness of the animations to the time slider and the speed-setting mechanism. With this change the amount of DOM manipulations necessary would be reduced, resulting in an expected performance gain. There is always going to be a limit to performance set by the total amount of objects to be animated. However, this limit is expected to be relatively high, as we observed that the large test-case of 4,825 occurrences did run fairly smoothly once the interactive setting of the parameters (and thus the DOM manipulation) was done.

14.5 Conclusion

We can conclude that combining distributed webservice and animated, interactive vector maps in a service-oriented, distributed architecture is possible with the use of existing interface, service and data standards with smaller files. We have built an OGC-compliant Web Map Service implementation that serializes spatio-temporal data from a database backend as animated Scalable Vector Graphics. The SVG is used in a web browser to show animated maps with a built-in advanced user-interface. This interface allows the user to interact with both the spatial and the temporal dimensions of the data.

A use case for the visualization of the movements of Antarctic icebergs was developed and our test-cases show that the system works as planned. There were clear performance problems in the interactivity, which makes this particular implementation less useful for exploratory purposes, where time-brushing and similar interactive features are needed. But, for less demanding tasks, where the speed of the animation can be fixed and brushing is not needed, the system could be very useful. In addition, we have identified the cause of the interactivity performance limitations, and feel we have pointers to solving them in the longer run. These results make us think that the TimeMapper system could be the starting point for a complete animated mapping system within an SDI context.

Our objective to design an animated mapping system which can generate SVG SMIL animations, automatically and directly, from spatio-temporal data, has thus been reached. Although our prototype only uses one specific data set, the employment of OGC standards and services in the system does make us confident that any other SDI-type of data could be used directly. It also works automatically, albeit in the more narrow sense explained in Sect. 14.1.1. The broader sense of fully automatic mapping from data, with cartographic design decisions included, would make a system “...simulate human action...” (from the more extensive definition in Simpson and Weiner (1989)). This still remains as an interesting research challenge that we hope to work on in the future.

References

- Andrienko N, Andrienko G (2006) Exploratory analysis of spatial and temporal data: a systematic approach. Springer, Berlin
- Andrienko N, Andrienko G, Gatalsky P (2003) Exploratory spatio-temporal visualization: an analytical review. *J Vis Lang Comput* 14:503–541
- Becker T (2009) Visualizing time series data using web map service time dimension and SVG interactive animation. MSc thesis, ITC, Enschede
- Behr F-J, Li H, Cheng H, Weldeslasie F, Chen Z (2006) PHPMyWMS – an open source based, SVG oriented framework for extended web map services. In: Wu H, Zhu Q (eds) *Geospatial information technology – proceedings of SPIE*, vol 6421. SPIE, Bellingham
- Benn DI, Warren CR, Mottram RH (2007) Calving processes and the dynamics of calving glaciers. *Earth Sci Rev* 82(3–4):143–179

- Blok CA (2005) Dynamic visualization variables in animation to support monitoring of spatial phenomena, vol 328, Netherlands Geographical Studies. KNAG/Universiteit Utrecht/ITC, Utrecht/Enschede
- Brinkhoff T (2007) Increasing the fitness of OGC-Compliant Web Map Services for the Web 2.0. In: Fabrikant S, Wachowicz M (eds) The European information society, Lecture notes in geoinformation and cartography. Springer, Berlin, pp 247–264
- de By RA, Lemmens R, Morales J (2009) A skeleton design theory for spatial data infrastructure. *Earth Sci Inf* 2(4):299–313
- EasyWMS (2011) Easywms open source webgis projects. URL: <http://www.easywms.com/>
- Fowler H, Fowler F, Sykes J (eds) (1976) Concise Oxford dictionary of current English, 6th edn. Clarendon, Oxford
- Harrower M, Fabrikant S (2008) The role of map animation for geographic visualization. In: Dodge M (ed) Geographic visualization: concepts tools and applications. Wiley, Chichester, pp 44–66
- ITC (2010a) RIMapper project pages. URL: <http://kartoweb.itc.nl/rimapper/>
- ITC (2010b) TimeMapper project pages. URL: <http://geoserver.itc.nl/timemapper/>
- Köbben B (2007) RIMapperWMS: a Web Map Service providing SVG maps with a built-in client. In: Fabrikant S, Wachowicz M (eds) The European information society – leading the way with geo-information, Lecture notes in geoinformation and cartography. Springer, Berlin, pp 217–230
- Köbben B, de By R, Foerster T, Huisman O, Lemmens R, Morales J (2010) Using the SDIlight approach in teaching a geoinformatics master. *Trans GIS* 14(s1):25–37
- Kraak MJ, Edsall R, MacEachren AM (1997) Cartographic animation and legends for temporal maps: exploration and/or interaction. In: Ottoson L (ed) Proceedings of the 18th ICA international cartographic conference, vol 1. International Cartographic Association, Stockholm, pp 253–260
- Li Y, Dong X, Chi G (2009) Towards spatial information service with applied extension to SVG. In: Proceedings of new trends in information and service science, 2009 – NISS 09, Beijing, pp 474–480
- National Ice Center (2010) Antarctic icebergs. URL: <http://www.natice.noaa.gov/products/iceberg/>
- OGC (1999) OpenGIS simple features specification for SQL revision 1.1. Technical Report OGC 99-049, Open Geospatial Consortium
- OGC (2002a) Implementations by specification. URL: <http://opengeospatial.org/resource/products/byspec/>
- OGC (2002b) Web map service implementation specification (1.1.1). Technical Report OGC 01-068r3, Open Geospatial Consortium
- OGC (2006) Web map server implementation specification 1.3.0. Technical Report 06-042, Open Geospatial Consortium
- Ostländer N, Tegtmeyer S, Foerster T (2005) Developing an SDI for time-variant and multi-lingual information dissemination and data distribution. In: Proceedings of 11th EC-GI & GIS Workshop ESDI: setting the framework, Alghero, Sardinia, European Commission
- Schmittner A, Yoshimori M, Weaver AJ (2002) Instability of glacial climate in a model of the ocean–atmosphere–cryosphere system. *Sci Exp* 295(5559):1489–1493
- Simpson J, Weiner E (eds) (1989) Oxford English dictionary, 2nd edn. Clarendon, Oxford
- W3C (2008) Synchronized multimedia integration language (SMIL 3.0). URL: <http://www.w3.org/TR/SMIL/>
- W3C (2010) Scalable vector graphics (SVG) 1.1, 2nd edn. URL: <http://www.w3.org/TR/SVG11/>