

SUAS MAPSERVER - AN OPEN SOURCE FRAMEWORK FOR EXTENDED WEB MAP SERVICES

Franz-Josef Behr^{a,*}, Hui Li^b

^aDepartment of Geomatics, Computer Science and Mathematics, University of Applied Sciences Stuttgart
Schellingstraße 24, D-70174 Stuttgart (Germany) - franz-josef.behr@hft-stuttgart.de

^bM-Way Solutions GmbH, Leitzstr. 45, D-70469 Stuttgart - leeglanz@gmail.com

Commission IV, WG IV/5

KEY WORDS: Internet/Web, Mapping, Services, SVG, Interoperability, Spatial Infrastructures, Standards, SVGT

ABSTRACT:

The Web Map Server Implementation Specification (WMS), originally developed and published by the Open Geospatial Consortium, was finally adopted by ISO as an international standard. According to this standard maps are presented either in “picture” formats or “graphic element” formats. While the pictorial (visual) representation uses raster formats, graphic element formats include Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) format.

In this paper a framework named SUAS for supporting extended Web Map Services will be presented. Geo-data is converted to Well Known Text format (WKT) and stored in a database management system, retrieved by the WMS server upon user request and transferred – optionally compressed – through the Internet. Finally it is visualized in Web browsers on desktop computers or mobile devices, natively or using special map clients. Besides supporting the WMS standard operations, additional formats and extensions have been included in this framework. The development proves that open, XML-based standards in combination with modern programming languages and integrated development environments allow rapid implementation of recommendations and standards in geo-informatics.

1. INTRODUCTION

1.1 General background

With large number of spatial data resources available, the Web Map Server (WMS) Implementation Specification, originally developed and published by the Open Geospatial Consortium (OGC 2002a, OGC 2006a) was finally adopted by ISO as an international standard titled “IS 19128:2005 Geographic information - Web map server interface”. It is a basic specification enabling users to transparently access data of interest from one or several map servers.

This standard is indeed a remarkable technical and commercial breakthrough in Internet mapping. Software conforming to this specification “*is able to automatically overlay map images obtained from multiple dissimilar map servers, regardless of map scale, projection, earth coordinate system or digital format*” (OGC 2004). Its success is based on the usage of a standard web protocol (HTTP), a standardized markup language (XML) and ordinary Web browsers. Request URLs may be bookmarked, pasted into HTML documents, and so forth, a benefit for the dissemination of WMS. This approach is also supported by the SUAS MapServer described here.

The service itself consists of three different operations with a corresponding set of parameters. As often defined for such services the first operation – the GetCapabilities request – provides general meta information to the user about available data, projection, and format, the so called service metadata. A client has to parse the XML capabilities document to retrieve the necessary information for the next step, the GetMap request.

Through this operation the map itself is obtained and can be visualized by the client software, sometimes combined with maps obtained from other servers. Such requests can be restricted to specific layers, size, extent and desired spatial reference system.

These two operations belong to the basic Web Map Service which can be extended to a queryable WMS by supporting a third operation, the GetFeatureInfo request which provides additional attributive information about geographical features in the map.

While textual output for service metadata, error messages (exception reports), or responses to GetFeatureInfo requests are usually formatted as Extensible Markup Language (XML), maps are presented either in picture formats or graphic element formats.

The pictorial (visual) representation using raster formats is most often implemented due to two reasons: The specification originally denotes a map as a visual representation of geo-data, not the data itself (OGC 2002a). Secondly, standardized raster formats such as Graphics Interchange Format (GIF), Portable Network Graphics (PNG), or Joint Photographics Expert Group (JPEG) can be displayed by common Web browsers, additional formats such as Tagged Image File Format (TIFF) can be visualized by additional software invoked by the Web browser.

For the representation in graphic element formats the specification mentions Scalable Vector Graphics (SVG, <http://www.w3.org/Graphics/SVG/>) or Web Computer Graphics Metafile (WebCGM, <http://www.w3.org/TR/REC-WebCGM/>).

* Corresponding author.

These formats constitute a scale-independent description of graphical elements, so that scale and size of the display may be modified while preserving the relative arrangement of the graphic elements, as ISO 19128 describes.

In this paper a framework for supporting Web Map Services with partial focus on SVG will be described. Geo-data is converted to WKT, stored in a ordinary relational database management system (DBMS), is retrieved and transferred – optionally compressed – through the Internet, and is finally visualized in Web browsers on desktop computers or mobile devices, natively or using plug-ins extending the browser's functionality.

1.2 Related work

A few approaches for open geospatial services are described in literature and Internet resources:

- A widely used web mapping application is *MapServer*, formerly called UMN MapServer due to the development by the University of Minnesota (UMN, see <http://mapserver.gis.umn.edu/>). This application framework is now maintained by developers around the world as part of the Open Source Geospatial (OSGeo) Foundation. MapServer supports several OGC standards, including WMS and Web Feature Service (WFS). Because it runs as an executable program, the installation however requires specific rights on server side which is not always given.
- *Deegree*, developed by lat/lon GmbH and the GIS Research Group of the Department of Geography of University of Bonn, is a Java Framework offering “the ... building blocks for Spatial Data Infrastructures” (<http://www.deegree.org>) by supporting a wide range of standards from OGC and ISO/TC 211 (like WMS, WFS, Web Coverage Service and Catalogue Service Web-Profile). However Java Servlet and Java Server Pages technologies are not provided by many Internet service providers; therefore SUAS is built on PHP and MySQL which are widely supported. Additionally experiences show that setup of Degree is not as easy as one could expect (Schweizer 2005, Pareeth 2008).
- *MapGuide Open Source*, sold until November 2005 as Autodesk MapGuide, is a web-based platform to develop and deploy web mapping applications and geospatial web services. Autodesk submitted this product to the OSGeo Foundation in March 2006 under the LGPL license (<http://mapguide.osgeo.org/>).
- Williams (2005a) and Neumann (2006) use PostGIS (<http://www.postgis.org/>) to store geometries in WKT format. A server-side scripting language (PHP) extracts data from the database and creates SVG geometry fragments on the fly supporting a service similar to WMS's GetMap request.

Besides these open source solutions several commercial internet mapping software packages exist like ArcGIS Server, GeoMedia Web Map, or MapXtreme; these approaches are not discussed.

1.3 Web Map Server related Specification

There exist several versions of WMS implementations on the market. While WMS 1.1.1 is implemented by more than 200 products (OGC 2008), the recent version, WMS 1.3, is supported only by 45 products. This demonstrates the use, popularity and stability of the version 1.1.1 specification. Also the SUAS framework described here is currently based on WMS 1.1.1 specification.

Styled Layer Descriptor (SLD, OG 2002) specification is designed to permit a wide variety of implementations of WMS which support user-defined symbolization. SUAS supports the SLD specification and can produce maps with user-defined symbolization of features instead of named Layers and Styles.

With the WMS operations DescribeLayer, GetLegendGraphic and GetStyles, a client can retrieve further information about features and styles from a server.

2. SCALABLE VECTOR GRAPHICS

2.1 Overview

SVG is a XML-based language for describing two-dimensional graphics like vector shapes, raster images and text. Graphical objects can be grouped (similar to layers in GIS), styled, transformed, animated and composed based on predefined objects. Additionally the SVG recommendation includes nested transformations, clipping paths, alpha masks, filter effects, template objects and extensibility (Watt 2001).

SVG 1.1 is a W3C Recommendation since 2003 and forms the core of current SVG developments. SVG 1.2 is the specification still under development as W3C Working Draft.

Because of industry demand, two mobile profiles were introduced with SVG 1.1: SVG Tiny (SVGT) and SVG Basic (SVGB, see W3C 2006). These subsets of the full SVG standard are mainly intended for user agents with limited capabilities like cellular phones or higher level mobile devices, such as PDAs. Therefore SVG Tiny has been used for the mobile client developed in this framework (cf. section 4.2).

SVG provides excellent graphics allowing magnification of an image on client-side without sacrificing sharpness, detail, or clarity or additional server requests. Title and description elements can be provided for supporting accessibility for people with disabilities (Campin 2003, Bulatov 2004). Screen readers can speak the title when objects are selected. For visually impaired users, a tactile copy of the SVG display can be created by printing to a tactile device which provides haptic effects.

SVG is especially suitable for applications that require continuous map updates and/or high level of interactivity at the client using AJAX technology where this format can be used besides other XML formats or JSON. Interactivity is supported in the browser using ECMAScript.

Supporting XML format, the server's output is not only a “portrayal of geographic information”, but can be used additionally to deliver the geo-data itself, comparable to the output of a Web Features Service (Kettemann & Zhu 2005). As an XML based grammar SVG enables combination with other

technologies like XLink and SMIL and communication between applications on different platforms.

2.2 Browser support

Display of SVG documents is supported by native browser implementations (Mozilla, FireFox, Opera, and Safari), plug-ins, stand-alone viewer applications like Batik (<http://xmlgraphics.apache.org/batik/>) and mobile devices. The mapping APIs of Google Maps and Windows Live Local use inline SVG in Firefox and Opera to render driving direction paths and other shapes which also indicates the importance of SVG.

Even if Adobe announced to discontinue the support for *Adobe SVG Viewer* on January 1, 2009, this viewer is still a popular browser plug-in available for Windows®, Mac, Red Hat and Solaris operating system, especially important for map display in Internet Explorer.

Two mobile viewers should be mentioned: *TinyLine SVG Toolkit* is a free, pure J2ME CLDC 1.0 implementation for visualization of SVG Tiny documents on mobile devices (<http://www.tinyline.com/svg/index.html>), available for Windows CE, Embedded Linux, PalmOS, and Symbian. This viewer is used in the mobile viewer of the WMS framework, as described below. It can be run on a phone without a built-in SVG implementation. *BitFlash SVGT Player*, a commercial product, available for different operating systems including Symbian and WindowsCE claims to be fully compliant with the W3C's SVG Tiny 1.1 and 1.2 specifications. (http://www.bitflash.com/prod_playerSVGT.html). Because of the many advantages of SVGT, more and more mobile phones begin to support this format; as mentioned at http://svg.org/special-svg_phones/, more than 160 SVG-enabled phones are currently available.

3. THE WMS SERVER IMPLEMENTATION

3.1 Database structure

The database – in this framework currently MySQL is used – consists mainly of two tables as described in table 1. The *FeatureClass* table stores meta-information about single layers. A *layer*, in this context, is a collection of features belonging to the same feature class. The features themselves are stored in the *FeatureGeometry* table.

FeatureClass	FeatureGeometry
fcid	id
stid	aid
layer	layer
description	geomtype
geomtype	srs
layertype	xmin, ymin, xmax, ymax
xmin, ymin, xmax, ymax	geom
srs	xlink
queryable, visible	textcontent
priority	attributes
elevation	
recnum	
size	

Table 1: Essential tables and attributes of the SUAS MapServer database

The attribute *layertype* describes where and in what format the features of this layer can be found. The attribute *description*

contains short textual information about the layer retrieved for the *GetCapabilities* request. The bounding box for the features is stored in the attributes *xmin*, *ymin*, *xmax*, *ymax* defining the lower left resp. the upper right corner. The spatial reference system is contained in the *srs* attribute usually specified according to the EPSG Geodetic Parameters dataset. *Stid* field points to the id of style for this collection, which Style properties for a feature can be supplied optionally. Raster data has no attributes and is not *queryable* when using *GetFeatureInfo* request, *queryable* field should be set to true. The field *elevation* is used for building 3D models.

The *FeatureGeometry* table partially consists of similar attributes, but is related to single features. The type of geometry (region, polyline. ...) is stored in the attribute *geomtype*, the geometry itself in Well Know Binary format (WKB) in the attribute *geom*. *xlink* can be used to embed the SVG geometry into a hyperlink according to W3C's XLink recommendation (<http://www.w3.org/XML/Linking>). The *attributes* field contains the feature's attribute / value pairs in XML format for ease of parsing on client-side. The field *textcontent* is used for storing the label of a geometry as text string.

3.2 Service Meta data

During installation the relevant administrative metadata according to OGC's WMS Capabilities DTD (OGC 2002a, Appendix A.1) have to be specified and are stored for further use as global variables of the PHP application. Additional metadata for the *GetCapabilities* response are derived from the *FeatureClass* table. For a *GetCapabilities* request the server generates a XML metadata document, which includes information about layers, coordinate systems, bounding box values for each layer, contact person, company etc. Additionally the image formats supported (see section 3.4) are listed in the format elements of the metadata. The extended capabilities and operations are also defined by instances derived of *_ExtendedCapabilities* or *_ExtendedOperations* elements. All extension names have been selected with care to avoid conflicting with other names mentioned in the standard.

3.3 Data loading

The WMS framework offers several methods to load spatial data into the spatial database. The first input method uses CSV formatted files (comma separated values) with information about layer name, extents, geometry, reference system and attributes of spatial features. Due to their simple but comprehensive structure, CSV files can be generated quite easily by many GI systems.

The second method uses SVG formatted files containing feature information within SVG elements. All geographic information such as element ID, layer name, shape type and shape style, can be picked up and stored into the database.

Additionally some widely used formats such as MapInfo Data Interchange Format (MID/MIF), ESRI's ArcGIS Shape format, and ArcInfo E00 format are supported for bulk-loading of geo-data. Consequently SUAS supports explicitly commonly used formats of commercial GI systems.

3.4 Server Side Rendering

The server's rendering component supports PNG, GIF, JPEG and WBMP raster image format using the built-in library GD of

PHP5, but also extends the WMS specification by providing SWF and PDF output.

For Shockwave Flash Format (SWF) an additional library named Ming is used. Ming is open-source and allows creating Flash movies, including almost all of Flash's features, like shapes, gradients, bitmaps etc (<http://de.php.net/ming>).

Because PHP5 does not include a built-in module for generating Portable Document Format (PDF), currently a library named PDFlib is used (<http://www.pdfli.com/>). Because the whole framework should rely upon open source it is aimed to replace it by PDFlib Lite which can be used for free for open source projects (<http://www.pdfli.com/purchase/license-lite.html>), FPDF (<http://php.scripsi.de/browse/package/421.html>) or PDF-PHP (<http://www.ros.co.nz/pdf/>). The output tends to be similar to the PDF created by Map2PDF (<http://www.terragotech.com/solutions/index.php>), but without ECMAScript-based behaviour.

The output formats SVG, GML, KML, and X3D are XML grammars; hence server responses are generated upon request. For this task the Expat Library is used, the underlying XML parser for the Mozilla project, Perl's XML::Parser, and other open-source XML Parsers.

Virtual Reality Modelling Language (VRML) has its own data structure. A corresponding PHP class has been written to create the 3D model from the WKT geometries. VRML data in complete structure will be outputted line by line.

For these additional output formats some restrictions exist. For PNG, PDF, SWF, GIF, JPEG and WBMP format, Bezier curves and elliptical arc are not supported. For SVG and SVGT all types of geometry can be used.

3.5 Extended operations

Two special operations extend the capability of map rendering: *GetMap25D* and *GetMap3D* (cf. Fig. 1). A 2.5D map has the strongpoint of 2D and 3D. It can have smaller file size than a raster image, but still has the 3D intuitionistic sense. Map in form of 2.5D has been widely used in mobile navigation system. These two operations follow part of the specification of Web 3D Service (W3DS), which is a portrayal service for 3D geodata. SUAS can extrude 2D geodata and coordinate system to 3D geodata and coordinate system with given elevation data, and produce a 3D scene graphs in X3D, VRML, KML, and GML format. Because it is "fake" 3D data, these two requests were named differently from the *GetScene* request in W3DS to avoid conflicting names.

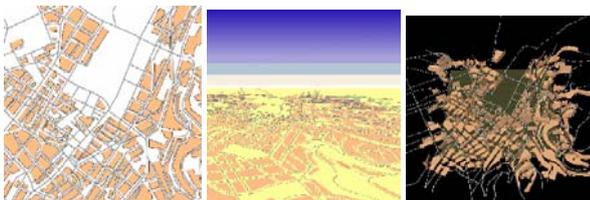


Figure 1: From left to right: 2D map from *GetMap* request, 2.5D map from *GetMap25D* request and 3D map from *GetMap3D* request.

As described above the *GetFeatureInfo* request enables the client to query attributes of features visible in the map.

Traditionally features are selected by layer name and pixel based coordinate values. Using SVG however as map output format, real-world coordinates can be used in the map displayed on client-side. By evaluation of ECMAScript events like *onclick*, these coordinates can be determined and integrated in an extended *GetFeatureInfo* request. In the same way a unique identifier can be derived through navigating in the SVG Document Object Model (DOM), as described by Williams (2005a). Such an identifier can also be used by SUAS MapServer's extended *GetFeatureInfo* operation for attribute retrieval.

3.6 User rights management

In order to support the protection of non-public geo-data SUAS MapServer embeds an additional user rights management system. Every user can upload his data and decide if it should be made publicly available or not. Though it is possible to establish a WMS platform as an open mapping community where users can share their geo-data. Each user can explore data from other users and integrate them into own *GetMap* requests.

4. THE WMS CLIENT IMPLEMENTATIONS

There are several WMS clients available on the market, but most of them are proprietary or work only for raster images without being able to truly interact with a WMS. One goal of this project is to implement map clients which can retrieve and analyze the data from WMS and especially support both raster and vector data such as SVG.

4.1 Browser based map client

A map client programmed in JavaScript, HTML and SVG allows the user to communicate with WMS servers for requesting maps. For the implementation of a dynamic WMS client, as essential technology Asynchronous JavaScript and XML (AJAX, Garrett 2005) is used to complete the asynchronous progress of data exchanging with servers in the background without experiencing of any visual interruptions.

The XMLHttpRequest object, the core technology of AJAX, is an optimal solution to execute the *GetCapability* request and fetch the responded XML file on client side. By using XML DOM parser, relevant elements of the XML document can be accessed, extracted and optionally modified. The element content is used for dynamic update of the user interface, i.e. to fill selection lists for layers and SRS, as well as for dynamic creation of internal data structures for preparing the *GetMap* request.

Finally based on the map data received from the server, different WMS widgets can be called to display the map itself. For SVG maps, a SVG based widget is started where the map display is controlled by several navigation tools, also created using SVG and JavaScript. This application is based on the work of Williams (2005a), Neumann et al. (2006) and Zhouer (2006).

For the display of raster maps, Openlayers (<http://openlayers.org/>) or other clients can be chosen as well to visualize and to explore the map. OpenLayers is a pure JavaScript library for displaying map data or building rich web-based geographic applications. The SUAS map client can use the Openlayers JavaScript API to interact with WMS.

4.2 Mobile map client

In this project a WMS client for mobile devices was designed and implemented (Li et al. 2006) based on SVGt and Java 2 Micro Edition (J2ME), a lean Java platform targeted specifically to clients like mobile phones, PDAs, and embedded devices.

The client developed has a connection management and provides a user interface derived from the service metadata (cf. Fig. 2). Not only SVGt data, but also PNG data can be displayed from GetMap and GetMap25D requests, and it has the capability to parse most of the GetCapabilities document responded from WMS, which follow the OGC WMS 1.1.1 implementation specification.



Figure 2: The mobile WMS client provides a graphical user interface; typical functions like zoom and pan are available for exploration of the map obtained from the WMS server.



Figure 3: Requirement checking in SUAS installation user interface.

4.3 Server installation

SUAS MapServer is available for download from <http://suas.easywms.com/>. In addition the project is hosted by SourceForge.net and mirrored by <http://www.gis-news.de/wms>. It offers flexible and friendly user interfaces for the ease of installation, configuration and maintenance as Figure 3 shows. Users do not need special knowledge to install the server and operate the complex database setting directly. SUAS provides an automatic installation procedure programmed in PHP and checks the availability of all installation requirements (see Table 2).

Apache HTTP server, version 2.0.x or 1.3.x
MySQL 5.16 or newer as database management system
PHP 5.05
GD library: for raster image, PNG, JPEG, GIF and WBMP generation.
MING library: for SWF image generation
PDF library: for PDF image generating
Expat library: for XML parsing using PHP
phpMyAdmin: to manage the database

Table 2: SUAS requirements.

5. CONCLUSIONS

An open source framework for a Web Map Service environment has been implemented to upload geo-data into a database and support the WMS operations GetCapabilities, GetMap, and GetFeatureInfo as well as extended operations by providing service metadata, map data in 2D and 3D and attribute information for selected features. Multiple formats including raster formats, SVG, SWF and PDF format as well as X3D/VRML, GML, and KML format are supported.

Due to its architecture and the software components used it can be installed and used on almost any web server. Map output can be displayed in standard browsers, SVG enabled browsers, and in two developed WMS viewer prototypes, one for ordinary browsers, another for mobile devices based on J2ME which can be ported to any Java-enabled phone.

However there are still some possible improvements:

- Some additional widely used formats such as AutoCAD's DXF or the GPS Exchange Format (GPX) should be supported for extension of file loading functionality.
- In addition to MySQL DBMS, SUAS should also support PostGIS, because it offers efficient spatial functions, such as buffering, overlaying and area selecting, which could improve the performance of SUAS.
- Further developments will support getting data from other WMS servers (while acting as a cascading WMS server) or creating map on the fly out of commonly used GIS data formats like Shape format.
- The WMS capabilities could be extended to handle efficiently tile requests as discussed by OGC 2007, a technology generally used by modern geo-browsers.

REFERENCES

- Bulatov, V., Gardner, J. A., 2004. Making Graphics Accessible. *Proceedings SVGOpen 2004*, Tokyo, Japan, <http://www.svgconference.com/2004/papers/SVGOpen2004MakingGraphicsAccessible/> [accessed 15.07.2006]
- Campin, B., 2004. SVG Maps for People with Visual Impairment. *Proceedings SVGOpen 2003*, Vancouver, Canada, <http://www.svgopen.org/2003/papers/svgmappingforpeoplewithvisualimpairments/index.html> [accessed 15.07.2006]
- Garrett, J. J. (2005). *AJAX: A New Approach to Web Applications*. Technical report, Adaptive Path.

- <http://www.adaptivepath.com/publications/essays/archives/000385.php>. Accessed on 09.4.2008.
- Li Hui., Behr, F.-J., Schröder, D., 2006. Design and Implement a Cartographic Client Application for Mobile Devices using SVG Tiny and J2ME. in: *Geoinformatics 2006: GNSS and Integrated Geospatial Applications*, edited by Deren Li, Linyuan Xia, Proceedings of SPIE Vol. 6418 (SPIE, Bellingham, WA, 2006) Article 641810
- Kettemann, R., Zhu, W., 2005. Vektordaten der Liegenschaftskarte „on demand“ im eigenen GIS, *Ingenieurblatt Baden-Württemberg*, Heft 2 / 2005
- Open Geospatial Consortium Inc., 2002. *Styled Layer Descriptor Implementation Specification*. http://portal.opengeospatial.org/files/?artifact_id=22364 [accessed 27.04.2008]
- Open Geospatial Consortium Inc., 2002a. *OpenGIS® Web Map Server Implementation Specification - Version: 1.1.1*. http://portal.opengeospatial.org/files/?artifact_id=1081&version=1&format=pdf. [accessed 15.07.2006]
- Open Geospatial Consortium Inc., 2004. *The OpenGIS Web Map Server Cookbook*. http://portal.opengeospatial.org/files/?artifact_id=7769 [accessed 15.07.2006]
- Open Geospatial Consortium Inc., 2006a. *OpenGIS® Web Map Server Implementation Specification - Version: 1.3.0*. http://portal.opengeospatial.org/files/?artifact_id=14416. [accessed 15.07.2006]
- Open Geospatial Consortium Inc., 2008. *Number of Specification or Interface Implementations*. <http://www.opengeospatial.org/resource/products/stats> [accessed 15.04.2008]
- Open Geospatial Consortium Inc., 2007. *OpenGIS® Tiled WMS Discussion Paper*. http://portal.opengeospatial.org/files/?artifact_id=23206 [accessed 29.04.2008]
- Pareeth, S., 2008. *OGC Compliant Solution for Online Digitisation Using Open Source Tools*. Unpublished Master's Thesis, University of Applied Sciences Stuttgart
- Rowley, T., Morris, J., Watt, J., Fritze, A., 2005. Implementing SVG in a Web Browser: Past, Present, and Future of Mozilla SVG. *Proceedings SVGopen 2005*, Enschede, Netherlands, <http://www.svgopen.org/2005/papers/MozillaSVG/index.html>, [accessed 15.04.2008]
- Schweizer, C., 2005. *Case study and realization of a web-based GIS Client-Server environment based on Open Source Products*. Unpublished Master's Thesis, University of Applied Sciences Stuttgart
- Watt, A., Lilley C. et al., 2001: *SVG Unleashed*. SAMS, 1117 S., ISBN 0-672-32429-6
- W3C, 2003. *Mobile SVG Profiles: SVG Tiny and SVG Basic*. <http://www.w3.org/TR/SVGMobile/> [accessed 20 Feb, 2006]
- Williams J. 2005a. Interactive Hiking Map of Yosemite National Park. *Proceedings SVGopen 2005*, Enschede, Netherlands, http://www.svgopen.org/2005/papers/abstract_williams_yosemite_national_park/index.html and <http://www.carto.net/williams/yosemite/> [accessed 15.04.2008]
- Williams, J., Neumann, A., 2005b. *Manipulating SVG Documents Using ECMAScript (Javascript) and DOM*. http://www.carto.net/papers/svg/manipulating_svg_with_dom_emascript/index.shtml [accessed 07.02.2005]
- Williams, J., Neumann, A., 2006. *Dynamic Loading of Vector Geodata for SVG Mapping Applications Using Postgis, PHP and getURL()/XMLHttpRequest()*. http://www.carto.net/papers/svg/postgis_geturl_xmlhttprequest/index.shtml [accessed 17.04.2008]
- Zhouer, Chen, 2006. *Design and Implement a Cartographic WMS Client Application using SVG and JavaScript*. Unpublished Master's Thesis, University of Applied Sciences Stuttgart