Chapter 3
Current Trends in Vector-Based Internet Mapping: A Technical Review

Christophe Lienert, Bernhard Jenny, Olaf Schnabel, and Lorenz Hurni

Abstract Possibilities and limitations of Internet cartography software largely depend on the pace set by the software industry. The variety of commercial and non-commercial software caters for the needs of a continuously growing mapping community, including both professional and amateur cartographers. This chapter provides an overview of state-of-the-art technologies for vector-based Web-mapping as of the beginning of 2011. Both proprietary and open format technologies are discussed for vector data rendering in browsers, highlighting their advantages and disadvantages. The discussed technologies are Adobe Flash, Microsoft Silverlight, Scalable Vector Graphics (SVG), JavaFX, Canvas, and WebGL. The chapter also discusses client and server side frameworks which provide Application Programming Interfaces (APIs) for creating custom interactive maps, mainly by overlaying raster images with vector data.

3.1 Introduction

Internet maps are the major form of spatial information delivery, as the Internet is today the primary medium for the transmission and dissemination of maps (Peterson 2008). For map authors, the maze of available techniques for creating and distributing Web maps is overwhelming, while authoring tools for Web-maps meeting the demands of high-quality cartography are difficult to find. Map authors may choose between GIS and graphics software products to create maps for the Internet, but these off-the-shelve maps oftentimes fall short of effectively conveying information. There are three main reasons for this shortcoming: (a) the design of these maps sometimes does not take into account the specific limitations of digital displays (Jenny et al., 2008); (b) the maps are often restricted in using standard

C. Lienert (✉)
Landscape and Waters, Canton of Aargau, 5001 Aarau, Switzerland
e-mail: christophe.lienert@ag.ch

M.P. Peterson (ed.), *Online Maps with APIs and WebServices*,
Lecture Notes in Geoinformation and Cartography,
DOI 10.1007/978-3-642-27485-5_3,

functionality provided by the authoring software; and (c) they do not take full advantage of interactive features available in modern Web-browsers. Not only is the situation of available products confusing and overwhelming, there are also out-of-date technologies, which are not developed further. Likewise, some of the new technologies are characterized by short life cycles: they have disappeared as fast as they have arrived on the scene.

In comparison to raster-based maps, vector graphics formats offer a series of advantages for interactive mapping: (a) They are scalable without loss of information or graphical artefacts; (b) the symbolization is adjustable on-the-fly (e.g., line width, transparency, fill color); (c) the geometry and symbolization can be animated; (d) map features can be shown and hidden without regenerating and reloading the entire map; (e) attributes can be attached to each individual map feature; (f) map features, such as diagrams, can be generated on-the-fly; and (g) the geometry can be changed, allowing for lossless projection to other coordinate systems (Schnabel and Hurni 2009).

In this chapter, we confine ourselves to the description and assessment of current technologies for vector-based mapping on the Internet. Raster-based Web-mapping is not part of this chapter, and only cross-platform and cross-browser technologies are treated. The chapter refers to the state-of-the-art as of the beginning of 2011 and the discussed technologies relate to the most current browser versions, i.e., Mozilla Firefox 4, Internet Explorer 9, Apple Safari 5, Opera 11, and Google Chrome 8.

3.2 Browser Technology for Vector Data Rendering

Vector mapping is based on vector graphics which use geometrical primitives such as points, lines, curves, or polygons. These primitives, in turn, are all based on mathematical equations. Base technologies for vector mapping may be defined as technology, or software, or even entire application programming interfaces (API), which are capable of creating, editing, and extending such vector-based graphics for the Web. The graphic objects may be changed by editing the geometry information and the graphical attributes. Affine transformation operators allow for stretching, twisting, and rotating the objects.

The most current and established vector technologies for the Web are presented in this section. Both the associated authoring tools and the way vector content is provided to the user are discussed. A distinction in proprietary and open-source software is made since considerable differences exist as to business and development models. An open-source approach allows for the extraction and further modification of vector graphics objects, or even for the technical advancement of authoring and rendering software. Proprietary source code, in contrast, is mostly delivered in a compiled binary form and is therefore non human-readable. Table 3.1 shows an overview of base technologies for vector mapping.

Java Applets and Vector Markup Language (VML) technology are not discussed in this article. Due to their complex programming environment, Java Applets are

Table 3.1 Base technologies for Web-based vector mapping

Technology/software	Company/consortium	Authoring tools	Format
Flash/Flex	Adobe	Flash Builder, Flash Professional	Proprietary
XAML/Silverlight	Microsoft	Expression Blend, Visual Studio	Proprietary
SVG	W3C	Illustrator, Corel Draw, Inkscape, XML Editors	Open
JavaFX	Oracle/Sun	NetBeans, Eclipse	Open
Canvas	WHATWG/W3C	–	Open
WebGL	Khronos Group	–	Open

comparatively little used to produce vector-based maps (Byrne et al., 2010). JavaFX is a modern alternative for the Java environment providing similar graphical capabilities, and discussed in this chapter. VML is deemed out-dated since it is a rejected World Wide Web Consortium (W3C) standard and is only supported by Internet Explorer (Zaslavsky 2003).

3.2.1 Proprietary Technology

The business models of software companies producing proprietary technologies and software are, by and large, based on licensing. Customers purchase a number of licenses which have to be renewed annually, or for each update. Usually, not the vector rendering technology itself is licensed, but auxiliary tools for creating the content.

Proprietary technologies and associated authoring tools are geared towards the designer community creating Web-based content, as well as programmers using various frameworks and code libraries for the development of Web applications. The two most widely used proprietary products are Adobe Flash and Microsoft Silverlight. They both provide high-performance authoring tools for graphic designers and programmers.

3.2.1.1 Adobe Flash

Originally developed by Macromedia, Adobe Flash was designed for animated Web-based vector graphics. Adobe's marketing targets graphic designers and authors of Rich Internet Applications (RIA). This orientation is reflected in the development of new tools, with Flash/Flex being the most well-known for application development (Noble and Anderson 2008). Currently, the cross-platform and cross-browser framework Flex comprises MXML (an XML-based vector graphics description language), ActionScript (a JavaScript-related language) and, for rendering, either the Flash browser plug-in, or the Adobe Integrated Runtime (AIR) for desktop applications. MXML is capable of describing various graphical user interface (GUI) components and vector objects. In addition, raster graphics, filter effects, videos, sound, animations etc. can be defined with MXML. User interaction can be realized with custom MXML ActionScript code.

Various authoring programs allow for the generation of Adobe Flash content. For designers, Adobe provides the visual design environment Flash Professional, while programmers draw on the tools from Adobe Flash Builder. In a typical workflow, either an MXML file is created using Adobe Flash Builder, or FLA files and/or ActionScript classes are created using Adobe Flash Professional. The resulting files are then compiled to a binary SWF file and presented with the Flash plug-in or the Adobe Integrated Runtime (AIR).

Advantages of the Flash framework include the performant rendering engine, the integration of multimedia content (e.g., video, sound and animation), a wide range of auxiliary tools for designers, and the wide-spread dissemination of the Flash Player for rendering Flash content. Adobe claims Flash Player is installed on more than 98% of Internet-enabled desktops worldwide (source: adobe.com/products/flashplayer/faq). Among the disadvantages of Adobe Flash, there are the dependency on one software vendor who may arbitrarily change the code base or the functionality of tools and plug-ins. Also, security concerns are raised when using a plug-in, particularly in regard to arbitrary, remote code execution and passing on of cached user information (Bradbury 2010).

Yet, Flash remains popular in the graphics industry. Typical use cases include games and multimedia graphics with animation or video, advertisement banners, and RIAs of varying complexity. Due to its wide dissemination, GIS and Web-mapping applications feature built-in map export functionalities compatible with Adobe Flash Player. An example is ESRI's ArcGIS API for Flex on top of the ArcGIS Server, which allows map authors to design customized interactive Web-maps, with options to edit or query data, and integrate temporal data.

3.2.1.2 Microsoft Silverlight

Microsoft's counterpart of Adobe Flash, the cross-platform and cross-browser Silverlight framework, consists of an XML-based vector graphics description language, known as XAML, which may be manipulated by various programming languages, such as C#, VB.NET, or JavaScript. Silverlight uses a subset of the Microsoft .NET framework, particularly the Windows Presentation Foundation (WPF). The necessary browser plug-in is available for Windows and Mac OS X and is installed on 50% of desktop computers worldwide (source: riastats.com).

Two authoring tools are available from Microsoft for generating Silverlight content: The visual authoring environment Microsoft Expression Blend for designers and the code-based Microsoft Visual Studio for programmers. In a typical workflow, a XAML file is created with Expression Blend or Visual Studio, compiled to a binary XAP file, and then presented in the browser by means of the Silverlight plug-in.

The performant rendering engine, the integration of multimedia content, and the availability of auxiliary tools for programmers are the main advantages of the Silverlight framework. The disadvantage in terms of the dependency on one single software vendor is similar to Adobe Flash.

Silverlight is suitable for programmers experienced with the Microsoft Windows .NET framework. It is supported by various development tools and Microsoft's dominant market position adds to its successful diffusion. Typical use cases include business applications. ESRI, traditionally closely connected to Microsoft, supports Silverlight with a separate API for creating interactive maps.

3.2.2 Open Standards

Open-source software is freely available and users may directly contribute to its enhancement by extending specific functionalities and publishing new code. In this section, four open-source technologies are discussed: Scalable Vector Graphic (SVG), Oracle/Sun JavaFX, WHATWG/W3C Canvas, and WebGL.

3.2.2.1 Scalable Vector Graphics (SVG)

SVG is a XML format for vector graphics. SVG is a recommended standard of the W3C consortium that all modern Web-browsers draw without the use of a plug-in, including Chrome, Firefox, Internet Explorer (as from version 9), Opera and Safari. However, the level of SVG support considerably varies between the different browsers. The SVG specification includes vector and raster graphics, filter effects, point symbols, masking, animation, and many other features (Neumann and Winter 2003). SVG is extendable with JavaScript, allowing for the creation of interactive graphics and graphical user interfaces.

Among the applications capable of creating SVG content are Adobe Illustrator, Inkscape, Xara Extreme, and Open Office Draw for designers, or different XML editors for programmers. In a typical workflow, an SVG file is created containing geometry data, and a separate JavaScript file with the application logic (e.g., the interactive functions).

The main advantages of SVG are the direct support in browsers, and the large variety of vector elements and visual effects. Another major advantage is the possibility to use multiple coordinate systems in a single drawing, which makes the SVG standard attractive for mapping applications: map features are based on native geographic coordinates, while user interface elements use screen coordinates.

The disadvantages of SVG are the sub-optimal support for multimedia, and the slow rendering. This issue is currently addressed by browser authors. Internet Explorer 9, for example, will introduce hardware accelerated SVG rendering.

3.2.2.2 JavaFX

JavaFX, now developed by Oracle, is a cross-platform and cross-browser framework for the development of Rich Internet Applications (RIA). It is based on the Java Runtime Environment, which is installed on about 75% of desktop computers worldwide (source: http://riastats.com). The tools needed for generating JavaFX content are NetBeans or Eclipse, both Integrated Development Environments (IDE) for experienced programmers. In a typical workflow, JavaFX code is compiled to Java bytecode and saved to a JNLP or JAR file. These files are then passed to the browser and executed using the Java Runtime Environment or Java Micro Edition on mobile phones.

Among the advantages of the JavaFX framework are the integration of Java drawing classes, and the thorough security concept. However, starting up the JavaFX plug-in is slower than starting up Silverlight or Flash. Another major disadvantages are missing tools for designers. Integrating multimedia elements, such as video and sound, is possible; but owing to the lack of authoring tools, JavaFX is mainly used by experienced programmers. Being an open-source framework with a thorough security concept, JavaFX comes into play for developing large business applications in which maps may be an integral part. However, for Web applications, it is currently not as widely used as is Flash or Silverlight. It should also be noted that with the release of JavaFX 2, the hitherto recommended JavaFX scripting language will not be developed any further.

3.2.2.3 Canvas

Canvas is a HTML element, which uses JavaScript commands for drawing graphic primitives (e.g., rectangles, paths, text). The Canvas version for drawing 2D graphics is standardized by the Web Hypertext Application Technology Working Group (WHATWG) and will be part of the upcoming HTLM5 specification. HTML5 is the next major revision of the HTML standard, which is currently under development by the W3C (Mansfield-Devine 2010). No browser plug-in is required to render Canvas elements, as it is already implemented in Chrome, Firefox, Opera, Safari, and Internet Explorer. Canvas is combinable with other Web standards, but it represents a lower conceptual protocol level than, for example, SVG, as it is not based on a built-in scene graph or a Document Object Model (DOM). Drawing commands are not converted to graphical features for later access or manipulation. Instead, each JavaScript drawing command immediately changes the pixels of the generated image. After rendering vector data, only the individual image pixels may be manipulated using JavaScript.

Currently, no mature graphic authoring tools exist for Web designers to create Canvas drawings. Content is therefore mainly created by programmers using text editors and custom-made code. In a typical workflow, a HTML file is extended with JavaScript code drawing the Canvas graphics. The JavaScript code might be embedded into the HTML file or stored in separate files. The browser automatically loads the JavaScript when rendering the Canvas element.

In the future, Canvas has a considerable potential to compete established vector data rendering technologies, such as Adobe Flash. The main advantages of the Canvas element are the support by all browsers, the fast rendering, and its options for raster data manipulation. The major disadvantage is the missing scene graph, which complicates linking with event handlers for interactive graphics, and which may considerably increase complexity when dealing with a large number of complex graphical primitives.

3.2.2.4 WebGL

WebGL is a 3D graphics API for Web applications that extends the HTML Canvas element. The specification is currently a work in progress, and implementations are not yet finalized. WebGL is specified by the non-profit technology consortium Khronos Group, which controls various open standards, for example, the OpenGL standard for rendering 3D graphics.

Similar to the 2D variant of Canvas, three-dimensional drawing with WebGL is controlled by JavaScript code without using a built-in scene graph. It accesses the computer's graphics card via the platform-independent OpenGL API which entails a very high rendering performance. WebGL uses the OpenGL ES 2.0 standard, a subset of OpenGL, which is also supported by devices with limited computing power, such as smartphones, tablet computers and other mobile devices. WebGL rendering is based on shader programs that calculate rendering effects on graphics hardware with a high degree of flexibility.

WebGL is not yet part of Web-browsers for end users. However, developer versions of Chrome, Firefox, and Safari contain experimental implementations. As a consequence, WebGL is currently mainly applied by programmers and early adopters for experimental applications.

Various scripting libraries are available to create WebGL content or for loading 3D objects that are designed with 3D modeling software (e.g., Autodesk 3ds Max). JSON (JavaScript Object Notation) is often used to describe and load 3D objects.

The advantages of WebGL include very fast rendering, and the versatility offered by shader programs for graphical special effects. Due to its early development status and owing to the lack of authoring tools, expert programmers are the exclusive user group of WebGL. Another disadvantage is the lack of support by Internet Explorer. However, WebGL has a considerable potential for both 2D and 3D map visualization.

3.3 Vector Overlay for Client-Side Mapping

In the previous section, independent general-purpose vector-based Internet standards are discussed. The standards are implemented in Web-browsers, or require additional plug-ins. The discussion is now moving towards a higher abstraction level, i.e., frameworks and APIs for mapping which build on these standards. Such client-side frameworks and interfaces offer additional functionality for cartographic applications, and encapsulate and further abstract the underlying visualization standards.

Client-side frameworks are widely used, since they greatly facilitate the creation of vector-based Web maps allowing cartographers to focus on their core competency in design and data visualization. The concept of such toolkits is to provide an API that allows map authors to create so-called mash-ups. Such maps usually combine a raster map in the background with custom, overlaid vector data. Often, the default graphical user interface provided by the API for manipulating the map is also customized, either by using functionality of the mapping framework, or by integrating specialized external libraries. The list below shows the most popular toolkits for generating map mash-ups.

The *Google Maps API*, the *Microsoft Virtual Earth/Bing Maps API,* and the *Yahoo! Maps API* offer similar functionality. They provide access to a multi-scale worldwide raster map, and a specialized graphical user interface for navigating the map. Authors work with a JavaScript based or a Flash-based API to embed their map contents. A wide range of functionalities and services are available for data integration and map drawing.

OpenLayers is a GUI and a customization tool for combining raster and vector data sources. It consists of a JavaScript library for displaying map data in Web-browsers, without any server-side dependencies. Unlike Google's, Microsoft's or Yahoo!'s APIs, the OpenLayers API is entirely free and open-source. It is often used in combination with OpenStreetMap, a freely editable map of the world.

The *carto.net* framework is for SVG based maps. By means of a programming language and the DOM (Document Object Model), SVG documents are manipulated. The DOM allows any Web-enabled programming language to create, manipulate, and delete map elements. In most cases, JavaScript is used for these manipulations.

CartoWeb (www.cartoweb.org) and *p.Mapper* are two frameworks running on MapServer, which is discussed in the next section. A graphical user interface and customization tools for web maps are provided, using JavaScript on the client-side and PHP MapScript on the server-side.

3.4 Vector Overlay for Server-Side Mapping

Web-map servers used to be restricted to raster-based output. Raster functionality includes the tiling of data, the conversion of data to various formats, or the resampling of raster images using different down- or up-scaling operators.

Nowadays, Web-map servers are increasingly able to produce vector graphics formats. The concept is much the same as for client-side mapping: vector data are handled as individual, addressable objects and often overlay background raster data. Different Web-map servers share common characteristics, such as their cross-platform and cross-browser capabilities, and their more or less strict support of Open Geospatial Consortium (OGC) standards. In this section, a selection of Web map servers are discussed in more details, with emphasis laid on vector formats.

3.4.1 MapServer

Formerly known as UMN MapServer, MapServer is the most widely used open-source map server worldwide. It is very popular with a large user community and with numerous programmers who further develop functionalities and features. MapServer is able to read data from a variety of enterprise geodatabases, such as Oracle, IBM DB2, or PostgreSQL via ESRI ArcSDE. It can also read data from spatial databases, such as Oracle Spatial, PostgreSQL/PostGIS, and from several GIS file formats, such as ESRI shapefiles. The main cartographic features include data filtering operations, anti-aliasing, on-the-fly projection, and visualization of data in form of pie and bar charts. Beside its capability to output raster data according to Web Map Service (WMS) versions 1.0, 1.1.1, 1.3.0, and Web Coverage Service (WCS) versions 1.0, 1.1.x, it also supports vector-based output standards, such as Geography Markup Language (GML), SVG, PDF, and Web Feature Service (WFS) version 1.0.0. In order to visualize vector output, the W3C standard Styled Layer Descriptor (SLD) 1.0.0 may be applied, but usually, users define the cartographic symbolization in a so-called mapfile. The advantages of MapServer are its active user community, on-the-fly map projection, and its easy integration in different Web servers, such as Apache and IIS. The fact that MapServer does currently not fully support SLD may be viewed as a disadvantage.

3.4.2 QGIS Mapserver

This open-source map server is based on Quantum GIS (QGIS), which is a free and open-source desktop GIS. It has a rather small, but very active user and developer community, mainly based in Europe. QGIS mapserver is able to read various data sources, ranging from ESRI Shapefiles to GML, or spatial databases like PostgreSQL/PostGIS. It features anti-aliasing and visualizes geodata by means of patterns, point symbols, or pie and bar charts. Beside the raster-based WMS 1.1.1 and 1.3.1 output, QGIS mapserver supports the vector-based output standards GML (Geography Markup Language), and WFS (Web Feature Service) in combination with SLD (Styled Layer Descriptor) 1.0.0. The SLD symbolization description file is typically generated using the Quantum GIS desktop application, or the associated PublishToWeb plug-in.

Among the advantages of QGIS mapserver are the fast rendering, and the possibility to visualize geodata as diagrams, patterns and point symbols, together with the full support for SLD. Another advantage is the integration in the Quantum GIS desktop application, making the functionalities of QGIS mapserver accessible to a wide user group. A disadvantage remains the small number of active developers.

3.4.3 GeoServer

GeoSever is an entirely Java-based open-source map server. Its worldwide community is large and active. GeoServer can handle data directly from most common spatial databases. PostgreSQL/PostGIS, IBM DB2 with Spatial Extender, Oracle Spatial, and ArcSDE, as well as standard GIS file formats such as ESRI Shapefiles are manipulable through GeoServer. Some of the advantageous features are the ability for anti-aliasing, versioning, and its security concept. Beside the most current raster-based standard outputs (see above for QGIS mapserver), GeoServer also exports to the vector formats WFS 1.0 and 1.1, PDF, SVG, KML, GeoRSS (geocoded Web feeds), as well as GML 2.1.2 and 3.1.1. It also fully supports SLD to create cartographic symbolization. GeoServer runs predominantly on the Apache Web server. The support by its active community and by major software companies is an additional advantage.

3.4.4 ESRI ArcGIS Server

This proprietary and popular map server allows users to link to the ESRI GIS software portfolio. Using ArcSDE, it handles spatial databases, such as Oracle Spatial, IBM DB2, and PostgreSQL/PostGIS, as well as a range of GIS file formats. Beside anti-aliasing, filtering, and 3D output, it also offers a variety of geo-processing functionality. ArcGIS Server is able to export raster-based data according to the most current standards, and also provides vector-based output, such as WFS 1.0 and 1.1, GML 2.0 and 3.1, KML 2.1 and 2.2. For cartographic symbolization, SLD 1.0 is supported. ESRI ArcGIS Server offers different Web-services, which are, however, typically conceptualized using the Desktop ArcGIS. The same applies to the definition of the cartographic symbolization. Among the major advantages are the geo-processing functionalities and 3D output.

3.4.5 *Intergraph Geomedia WebMap*

This software is another proprietary map server which handles spatial databases such as Oracle Spatial and Microsoft SQL Server. Geomedia WebMap is able to export WMS and different raster formats, as well as vector-based standards, such as WFS, GML, or SVG. Geomedia WebMap is mainly used in the business sector.

3.5 Web-Based Vector Map Examples

The following two map examples illustrate how geographic data may be visualized by vector-based Internet maps. The example in Fig. 3.1 contains hydrological real-time data which are automatically edited, processed, and visualized in an interactive vector map, along with interactive time series graphs. The map is based on data stored in a real-time PostgreSQL/PostGIS database which are automatically converted to SVG. The real-time visualizations have been created on the basis of the carto.net SVG framework and have been extended with specific interactive GUI components.

The example in Fig. 3.2 shows a city plan accessible to administrative officials as well as to the general public. The map is based on Microsoft Silverlight technology. Most of its content is delivered as WMS raster data using the ArcGIS Server via a

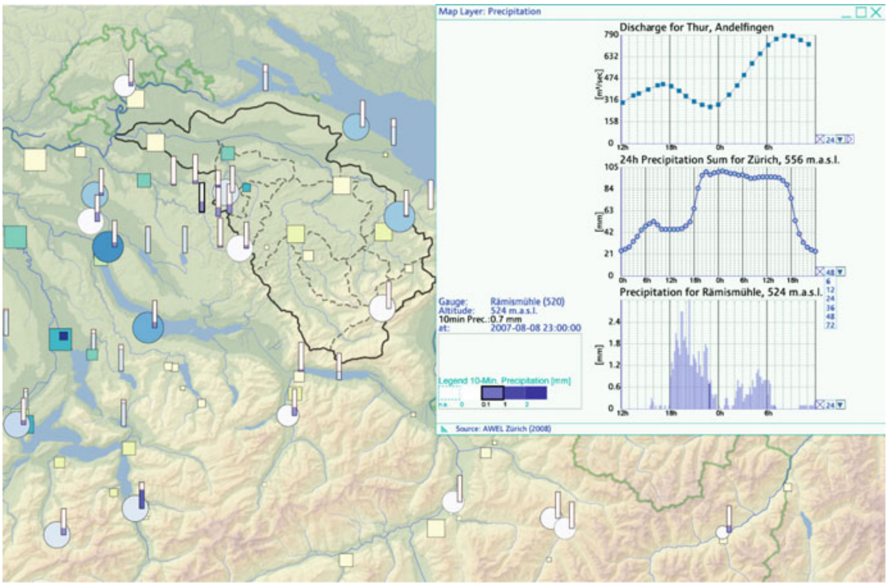

Fig. 3.1 SVG real-time map and time series graphs (Lienert et al., 2010)

Fig. 3.2 Silverlight city map (www.stadtplan.stadt-zuerich.ch)

REST API and integrated in a Silverlight GUI. Vector-based interactive objects, such as points of interest and borders, are placed over the raster background. The vector data in this example are stored in an Oracle Spatial database.

3.6 Conclusions

Vector-based Internet technologies are continuously developing and changing, with new standards and formats appearing, and old ones disappearing. The number of technologies may seem confusing, but at a technical level, similarities between the different technologies prevail: the use of equal graphical primitives (e.g., SVG and Silverlight), the integration of multimedia (e.g., Flash and Silverlight), and the use of JavaScript, or derivates thereof, outweigh the differences between the technologies, which are mainly rendering speed and the underlying authoring and programming environments.

The anticipated move from desktop applications to purely Web-based applications and services may drive browser vendors to further adopt Web-mapping standards and workflows (Jolma et al., 2008). Currently with self-contained mobile applications (so-called Apps) on the rise, however, the Web-browsers are not the ubiquitous user interface for which many were hoping (Kennard and Leaney 2010). Thus, a standardization of formats for vector-based Internet graphics is not to be expected soon.

Table 3.2 shows application domains and the disseminations of vector-based Internet mapping technologies by beginning of 2011, for different user groups and for different use cases. This table is surely subject to rapid change. A wide range of

Table 3.2 Application and dissemination of vector-based Internet mapping technologies

Technology/ software	Targeted authors	Typical use cases	Authoring tools	Dissemination
Flash/Flex	Designers and programmers	Multimedia, advertisement, games	++	Very high
XAML/ Silverlight	Designers and programmers	Business applications	++	Medium
SVG	Designers and programmers	Infographics (diagrams, maps)	+	High
JavaFX	Programmers	Business applications	+	High
Canvas	Programmers	RIAs, infographics, games	–	High
WebGL	Programmers	3D graphics (experimental)	–	Low

mapping frameworks and authoring tools cover various skill levels, some straightforward, others guided, yet others still experimental. However, the cartographers' choice of vector mapping technology certainly also correlates to their technical and programming skills.

Since authoring tools facilitate the map making process, technologies such as Adobe Flash and Flex, or Microsoft Silverlight may better meet the needs of design-oriented cartographers. Their popularity and diffusion is accordingly high. Due to their code-based environment and required programming skill, the features offered by software applications such as JavaFX, Canvas or WebGL reach a smaller number of cartographers. These technologies, however, enjoy growing popularity and may be complemented by some authoring tools in the future.

Client-side and server-side frameworks allowing for vector data overlay are becoming increasingly popular and are found on numerous commercial and non-commercial websites. There is also a trend for mapping software on mobile devices, such as cell phones, to adopt this overlay concept. For many map authors, client-side frameworks constitute a widely used basis for building customized maps. These frameworks provide free raster-based, multi-scale background world maps as well as functionality for custom vector data overlays. Technically more adept cartographers with more computer science expertise, in turn, may set up a Web map server for generating custom base maps combined with vector data overlays.

References

Bradbury D (2010) The battle of the Internet browsers. J Infosecurity 7(2):34–37

Byrne J, Heavey C, Byrne PJ (2010) A review of Web-based simulation and supporting tools. J Simul Model Pract Theory 18(3):253–276

Jenny B, Jenny H, Räber S (2008) Map design for the Internet. In: Peterson MP (ed) International perspectives on maps and the Internet. Springer, Berlin, pp 31–48

Jolma A, Ames DP, Horning N, Mitasova H, Neteler M, Racicot A, Sutton T (2008) Free and open source geospatial tools for environmental modelling and management. In: Jakeman AJ, Voinov AA, Rizzoli AE, Chen SH (eds) Developments in integrated environmental assessment. Elsevier, Amsterdam, pp 163–180

Kennard R, Leaney J (2010) Towards a general purpose architecture for UI generation. J Syst Software 83(10):1896–1906

Lienert C, Kunz M, Weingartner R, Hurni L (2010) Monitoring and comparing: a cartographic web application for real-time visualization of hydrological data. In: Konecny M, Zlatanova S, Bandrova TL (eds) Geographic information and cartography for risk and crisis management. Springer, Berlin, pp 409–424

Mansfield-Devine S (2010) Divide and conquer: the threats posed by hybrid apps and HTML 5. J Netw Security 2010(3):4–6

Neumann A, Winter AM (2003) Webmapping with Scalable Vector Graphics (SVG): delivering the promise of high quality and interactive web maps. In: Peterson MP (ed) Maps and the Internet. Elsevier, Amsterdam, pp 197–220

Noble J, Anderson T (2008) Flex 3 cookbook. O'Reilly, Sebastopol

Peterson MP (2008) International perspectives on maps and the Internet. Springer, Berlin

Schnabel O, Hurni L (2009) Cartographic web applications – developments and trends. In: Proceedings of the 24th international cartography conference, Santiago, 2009

Zaslavsky I (2003) Online cartography with XML. In: Peterson MP (ed) Maps and the Internet. Elsevier, Amsterdam, pp 171–196