

## GIS-RELATED WEB ENGINEERING AS TOPIC AND TOOL IN E-LEARNING

Th. Brinkhoff, B. Garrelts

Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven (University of Applied Sciences),  
Institute for Applied Photogrammetry and Geoinformatics (IAPG), 26121 Oldenburg, Germany -  
(thomas.brinkhoff, bruno.garrelts)@fh-oow.de

Commission VI, WG VI/1-2

**KEY WORDS:** e-learning, geoinformatics, web engineering, web services, earth viewers, AJAX, SVG

### ABSTRACT:

In the last few years, map viewers like Google Maps, Microsoft Live Search Maps and Open Street Map as well as virtual earth viewers like Google Earth and Microsoft Virtual Earth had an important impact on the development and dissemination of Internet-based GIS applications. Therefore, there is a vital need for knowledge about the techniques required for exploiting and developing such applications. The e-learning module "Web Engineering in the Geoinformatics" offers an unbiased introduction into this field. The module was developed within the current project "FerGI+" (Fernstudienmaterialien Geoinformatik; English: distance learning material for geoinformatics) for teaching students of geoinformatics, geodesy, geography and environmental sciences in a blended learning scenario as well as for participants of further education programs. It covers technologies like the Google Maps API, AJAX (Asynchronous JavaScript and XML), the OpenLayers API, OGC KML and SVG (Scalable Vector Graphics). The e-learning material is provided via HTML and a Moodle e-learning platform. The use of HTML allows a direct integration and visualization of the presented technologies into the e-learning material. The code required for the integration of a Google Maps or an OpenLayers application into a web page can be presented together with the resulting live map. An effective learning process demands changing between gaining new competences and using them in exercises in a high frequency. For this purpose, the students can modify given JavaScript functions within the HTML pages in order to work out the presented content. If it is done correctly, the resulting maps or SVG images are immediately visualized.

### 1. INTRODUCTION

In the recent years, providing geospatial information via Internet and World Wide Web (WWW) became increasingly important. Most popular are proprietary, but free-of-charge online *map services* like Google Maps and Microsoft Live Search Maps as well as *earth viewers* like Google Maps and Microsoft Virtual Earth. They cover the whole earth's surface by street and topographic maps as well as by aerial and satellite images. On top of these services, many so called *mashups* were implemented, i.e., web sites that seamlessly combine content from more than one source into a single integrated application. Following this development, projects of the open-source community gained attention: Open Street Map and OpenLayers are prominent examples.

One important consequence of this development is a vital need for knowledge about the techniques required for exploiting and developing such web applications. For this purpose, the new e-learning module "Web Engineering in the Geoinformatics" offers an introduction into this field. It covers technologies like the Google Maps API, AJAX (Asynchronous JavaScript and XML), the OpenLayers API, OGC KML and SVG (Scalable Vector Graphics). In addition of being subject of the module, many of these technologies are used for the development of interactive e-learning material. In other words, they are used as topic and as tool in e-learning.

The rest of the paper is organized as follows: After a short presentation of the FerGI+ project the main contents of the module are presented. The fourth section discusses how the outlined techniques can be used as tool for visualization

purposes and for supporting modifications of exemplary source code. The paper concludes with a short summary and an outlook to future work.

### 2. THE FERGI+-PROJECT

The project "FerGI+" started in April 2007 and was finished in March 2009 as a cooperative project of the University of Osnabrück and the Universities of Applied Sciences in Oldenburg and Osnabrück (Grendus et al., 2009). It was funded by the Ministry for Science and Culture of Lower Saxony during the third phase of the "eLearning Academic Network Niedersachsen" (ELAN III). During those two years, ten eLearning modules were developed. "FerGI+" is the successor of the well-known "FerGI" project (Schiewe et al., 2006); it broadens the range of offered e-learning modules. The contents of the modules are given in German and / or in English. An overview of the available FerGI(+) modules is given in figure 1. More information about FerGI+ can be found at <http://www.fergi-online.de>.

The FerGI(+) modules do not reflect the whole GI-curriculum, but concentrate on special GI-topics with high topicality. They are primarily used in blended-learning scenarios at universities, i.e. by a combination of long-distance learning and face-to-face learning. In addition, the modules are offered to all interested parties from administration, industry, schools and universities. An eight-week free-of-charge access to the modules is provided after a registration. Also special FerGI-based further education program exists for small companies that certificates successful attendance (Grendus, 2008).



Figure 1. Overview of the FerGI (without star) and FerGI+ modules (with star)

The Institute for Applied Photogrammetry and Geoinformatics (IAPG) of the University of Applied Sciences in Oldenburg developed five modules during the FerGI project (Krüger & Brinkhoff, 2005). For FerGI+, the three following modules were implemented:

- “Computational Geometry in the Geoinformatics”,
- “Data Structures and Algorithms for Spatial Database Systems”, and
- “Web Engineering in the Geoinformatics”.

The modules are provided via the WWW using simple and widespread techniques like HTML, CSS, JavaScript, PHP and Flash. This allows a simple integration of the modules into the technical infrastructure and the preferred (or forced) e-learning platforms of the universities like Moodle or ILIAS. In addition, a pure HTML version without the need of using an e-learning platform is offered.

### 3. CONTENTS OF THE MODULE “WEB ENGINEERING IN THE GEOINFORMATICS”

The overall topic of the e-learning module „Web Engineering in the Geoinformatics“ is the creation of web-based map applications by using proprietary and free-of-charge online map services. It covers also data formats like OGC KML and SVG and web technologies like AJAX that are relevant for the development of such applications. These topics are especially suitable for e-learning because they do not need only to be presented in an abstract way but can be directly exemplified by using the presented techniques.

#### 3.1 Earth Viewers

Online map services, which provide beside data (like maps and images) the required user control elements (like navigation controls and layer controls), are often called *earth viewers*. Typically, *web mapping systems* and *3D clients* are distinguished. Whereas web mapping systems can be used within ordinary web browsers, 3D client systems require the installation of additional software on the user's local computer. They are typically offering a three-dimensional view of the

earth as *virtual globe*. Both types of earth viewers provide the data on demand using streaming technology.

Among the most important providers of earth viewers are Google (Google Maps and Google Earth), Microsoft (Live Search Maps) and the NASA (World Wind). Their competition leads to a steady extension of the functionality and improvement of the usability of the viewers as well as to maps, images and photos of increased resolution and coverage.

A very important feature of the services mentioned above are their open *application programming interfaces (APIs)*. They allow incorporating the maps, images and user control elements into other web pages. Furthermore, it is possible to overlay these data by own geospatial data or other datasets loaded from the Internet. Most of the current so-called mashups are based on the API of Google Maps (ProgrammableWeb, 2009).

The module „Web Engineering in the Geoinformatics“ starts with an overview about existing online map services like Google Earth (see figure 2), Microsoft Live Search Maps / Virtual Earth API (figure 3), NASA World Wind and Open Street Map. Then, it presents two services in more detail: Google Maps and OpenLayers.

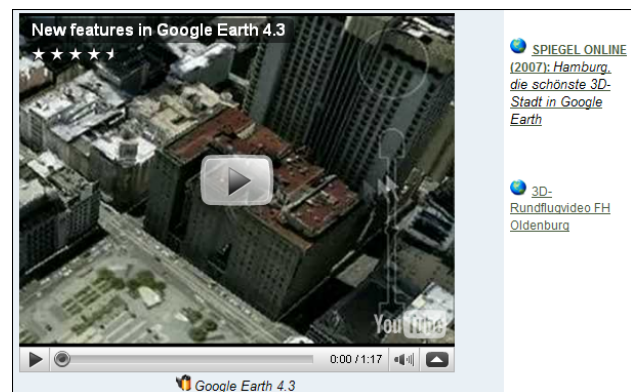


Figure 2. Integrated video about Google Earth and web links to articles and examples

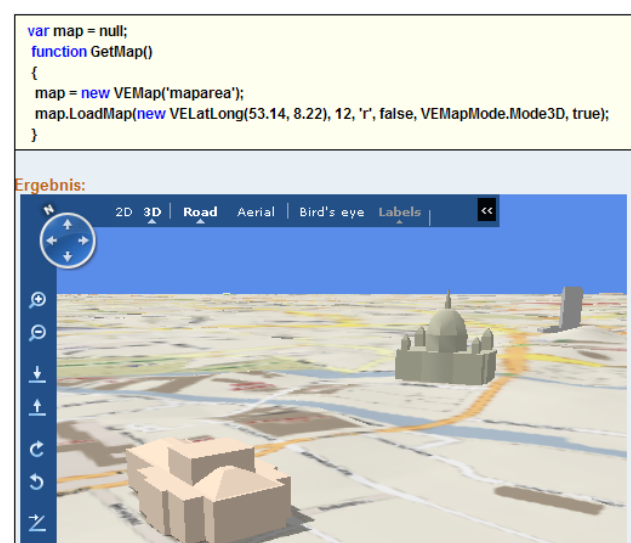


Figure 3. Explanation of the use of the Virtual Earth API and the result

**3.1.1 Google Maps:** In case of *Google Maps*, especially its API is explained (Google, 2009). The Google Maps API is a JavaScript library, which can be used with an individual API key requiring a free-of-charge registration at Google. The use of different map types, of user controls, of (user-defined and WMS) overlays and of geo-coding are outlined by the module (see figure 4). Like other proprietary map services, the Google Maps API supports only the use of Google-specific maps and images as background map.



Figure 4. An animation that illustrates the different layers of a Google Maps application

**3.1.2 OpenLayers:** *OpenLayers* is a free JavaScript library that allows integrating multiple map services into a web page (OpenLayers, 2009). It consequently separates the map controls from the map data. Therefore, the controls can be used for different data sources. Beside OGC-compliant web services, proprietary services, e.g., of Google, Microsoft and Yahoo and vector data and markers can be integrated. The module introduces the OpenLayers API and especially discusses the use of control elements, of different layer classes and the combination of proprietary and OGC-compliant services (see figure 5).

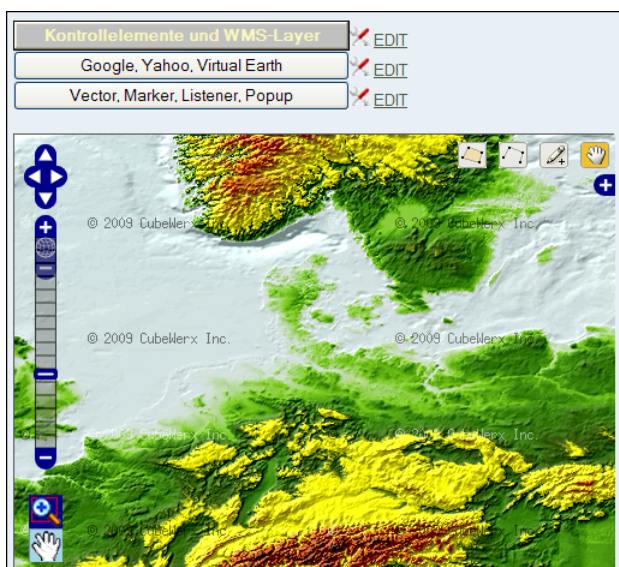


Figure 5. Exemplary OpenLayers application illustrating the integration of different data sources

## 3.2 Data Standards

The module „Web Engineering in the Geoinformatics“ covers two important data formats for geospatial web applications: OGC KML and SVG.

**3.2.1 OGC KML:** *OGC KML* (v2.2) is the successor of the KML 2.1 specification of Google (Wilson, 2008). It is a combined representation and visualization format for simple geometries. The e-learning module describes the data model for geometries, geospatial features and their styling (see figure 6).

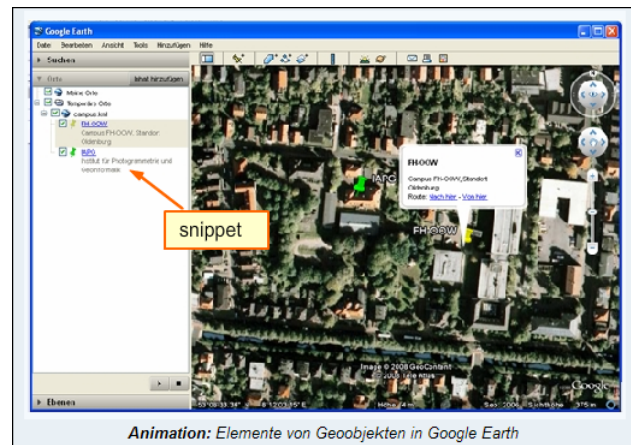


Figure 6. An animation that illustrates the elements of OGC KML features

|                                                                                                                                                                                                                                                                                                                                                                                        |                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T (absolut)<br>t (relativ)                                                                                                                                                                                                                                                                                                                                                             | (x y)+                                                 | <b>Shorthand/smooth quadratic Bézier curveto:</b><br>zeichnet eine quadratische Bézierkurve vom aktuellen Punkt zu der gegebenen (x,y)-Koordinate. Als Kontrollpunkt wird die Reflexion (bezüglich des aktuellen Punktes) des Kontrollpunktes der vorherigen Kurve verwendet.                                                                                                                                         |
| A (absolut)<br>a (relativ)                                                                                                                                                                                                                                                                                                                                                             | (rx ry x-axis-rotation large-arc-flag sweep-flag x y)+ | <b>elliptical arc:</b> zeichnet einen elliptischen Bogen vom aktuellen Punkt zu der gegebenen (x,y)-Koordinate. Die Radien der Ellipse werden über rx und ry angegeben. Über x-axis-rotation wird die Drehung der Ellipse im aktuellen Koordinatensystem definiert. Mit large-arc-flag (größere Bogen?) und sweep-flag (obere oder untere Bogen?) wird eindeutig bestimmt, welcher Bogen der Ellipse gezeichnet wird. |
| <b>Beispiele:</b>                                                                                                                                                                                                                                                                                                                                                                      |                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <pre>&lt;path d="M50 70 L150 70 L100 200 z" fill="red" stroke="blue"&gt; &lt;path d="M50 70 Q100 50 150 70 Q100 140 100 200 Q100 140 50 70" fill="green" stroke="yellow" transform="translate(120)"&gt; &lt;path d="M340 70 A40 40 0 0 1 340 200 A40 40 0 0 1 340 70 z M340 100 A20 20 0 0 1 340 170 A20 20 0 0 1 340 100 z" fill-rule="evenodd" fill="blue" stroke="orange"&gt;</pre> |                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Resultat:</b> <a href="#">EDIT</a>                                                                                                                                                                                                                                                                                                                                                  |                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                       |
|                                                                                                                                                                                                                                                                                                                                                                                        |                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                       |

Figure 7. Explanation, example and illustration of the SVG path element

**3.2.2 SVG: Scalable Vector Graphics (SVG)** (W3C, 2003) is an XML-based data format for two-dimensional vector graphics. It was developed by the WWW consortium especially for web applications. SVG is popular for displaying maps (Ueberschär & Winter, 2006), e.g., it can be the result format of WMS services. There are several reasons for its popularity for geospatial applications: SVG supports scaling, coordinate systems, and transformations. Furthermore, the document object model (DOM) of SVG allows accessing and modifying SVG elements and their attributes for implementing interactive map applications.

The e-learning module (see also figure 7) introduces the structure of SVG documents, coordinate systems, simple and complex geometry types, text and image elements, as well as the styling of those elements. More advanced topics are animations, scripting, the handling of mouse events and container elements.

### 3.3 Web Technologies: AJAX

One of the most important web technologies for interactive web applications is *AJAX* (Asynchronous JavaScript and XML). It is a technique for the asynchronous communication between a web server and a client application executed in a web browser. The communication is performed via the HTTP protocol. HTTP is a stateless protocol, i.e. the web browser expects for each request a response that is displayed by the browser. In case of interactive web applications this behavior leads to undesirable blockades and refreshing of displayed pages. AJAX bundles a clutch of techniques that allow performing an HTTP request and processing its result asynchronously without a complete reloading of the current page. AJAX allows the development of web applications that behaves similarly like traditional desktop applications. In case of a map application, e.g., spatial data can be requested subsequently and integrated into an already displayed map.

In the module „Web Engineering in the Geoinformatics“, the concept of AJAX is outlined as stand-alone solution as well as integrated into the Google Maps API (Purvis et al., 2006) (figure 8).

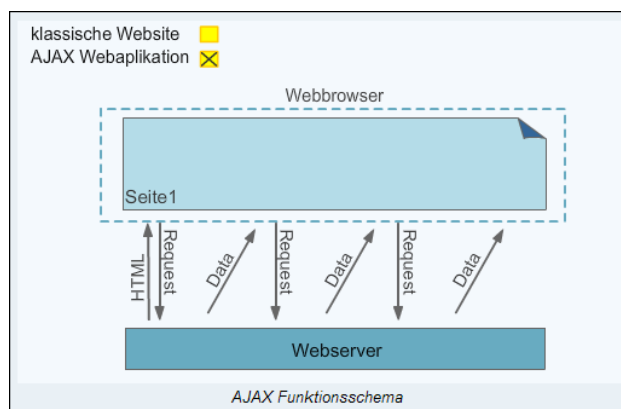


Figure 8. Animation that illustrates the functioning of traditional web requests and of AJAX

## 4. GIS-RELATED WEB ENGINEERING TECHNIQUES AS TOOL

The FerGI+ modules are realized by HTML. Therefore, a direct integration and visualization of the presented technologies into the e-learning material is possible. For example, the code required for the integration of a Google Maps or an OpenLayers application into a web page can be presented together with the resulting live map. Figures 3 and 5 illustrate such examples.

Furthermore, the students should be enabled to apply more examples directly and see the results. In the context of a web browser, this can easily be achieved by prepared links and calls of prepared JavaScript functions. The three upper buttons in figure 5 are examples for such an approach.

An effective learning process demands changing between gaining new competences and using them in exercises in a high frequency (Strobl, 2004). Therefore, the students should apply their new knowledge in modified contexts, e.g., for changed requirements or objectives. For this purpose, the students are enabled to modify given JavaScript functions or to create new code within the presented HTML pages. For a suitable feedback, the modified or new code will be evaluated straight away. In our context that means – if the programming is done correctly – the resulting map (or SVG image) is immediately visualized using the given JavaScript libraries and available map services.

All three variants of integration of the web engineering technologies are used in the chapters about the Google Maps API, about the OpenLayers API and about SVG.

### 4.1 Integration of the APIs for an Exemplary Visualization

The integration of the APIs for exemplary visualization purposes is done in all cases similarly by adapting or providing three different documents: the containing HTML page, a controlling JavaScript document and a CSS document for styling purposes.

The HTML document is added by (see also figure 9):

- a reference to the required JavaScript libraries (plus a registration key if necessary),
- an initialization of the service (typically by calling the initialization function via the `onLoad` event handler), and
- a container element for the map (typically a `div` element that is identified by an ID).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Google Maps</title>
<link href="style.css" rel="stylesheet" type="text/css"/>

<script src=
"http://maps.google.com/maps?file=api&v=2&key=XXX"
type="text/javascript"></script> // The library reference
<script src="script.js" type="text/javascript"></script>

</head>
<body onload="load()" // The initialization
<h1>The Map</h1>
<div id="maparea"></div> // The map container
</body>
</html>
```

Figure 9. The augmented HTML document in case of the Google Maps API

The JavaScript document is the component that controls the API functionality. There is typically a central class representing the map. Its constructor is parameterized by the ID of the container in the HTML page that should accommodate the map (figure 10).

```
var map = null;
var start_y = 53.13884;
var start_x = 8.214575;
var start_zoomlevel = 12;

// Initialization
function load() {
    if (!GBrowsersCompatible()) {
        alert("The browser is not compatible for Google Maps!");
        return;
    }
    // Create map object
    map = new GMap2(document.getElementById("maparea"));
    // Define the center of the map
    map.setCenter(new GLatLng(start_y, start_x, start_zoomlevel);
}
```

Figure 10. The controlling JavaScript document

## 4.2 The Support of Code Modifications

For modifying exemplary code, a corresponding *code editor* was developed. In contrast to the code fragments depicted within the learning material, the editor requires the complete code of an example. The editor is called by “EDIT” links that reference a corresponding code file; the figures 5 and 7 depict such links. Triggering such link opens the editor and loads the corresponding code into the included input area. Modifications are directly evaluated by the JavaScript interpreter of the web browser. The corresponding results (or error messages if case of incorrect code) are immediately visualized. Figure 12 illustrates this process.

The editor is realized by using HTML frames. That allows the user to vary the size of the input area and of the result area. Figure 11 shows the functionality of the editor.

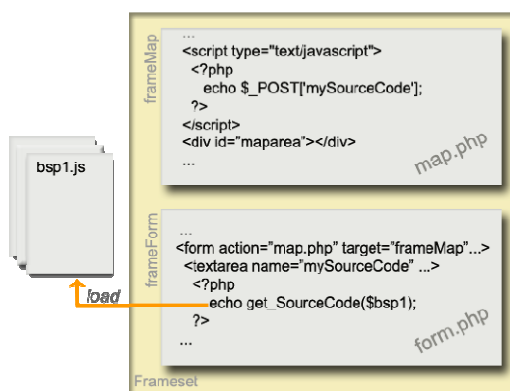


Figure 11. Functionality of the editor in case of maps

The same approach can be applied for the editing and the visualisation of SVG documents. In contrast to the map APIs, the upper area of the editor is not an HTML file but an XML file that can be directly evaluated by the web browser (if it has a native SVG support like Mozilla Firefox) or by a plug-in of the

web browser (like in the case of Microsoft's Internet Explorer). Figure 13 illustrates this approach.

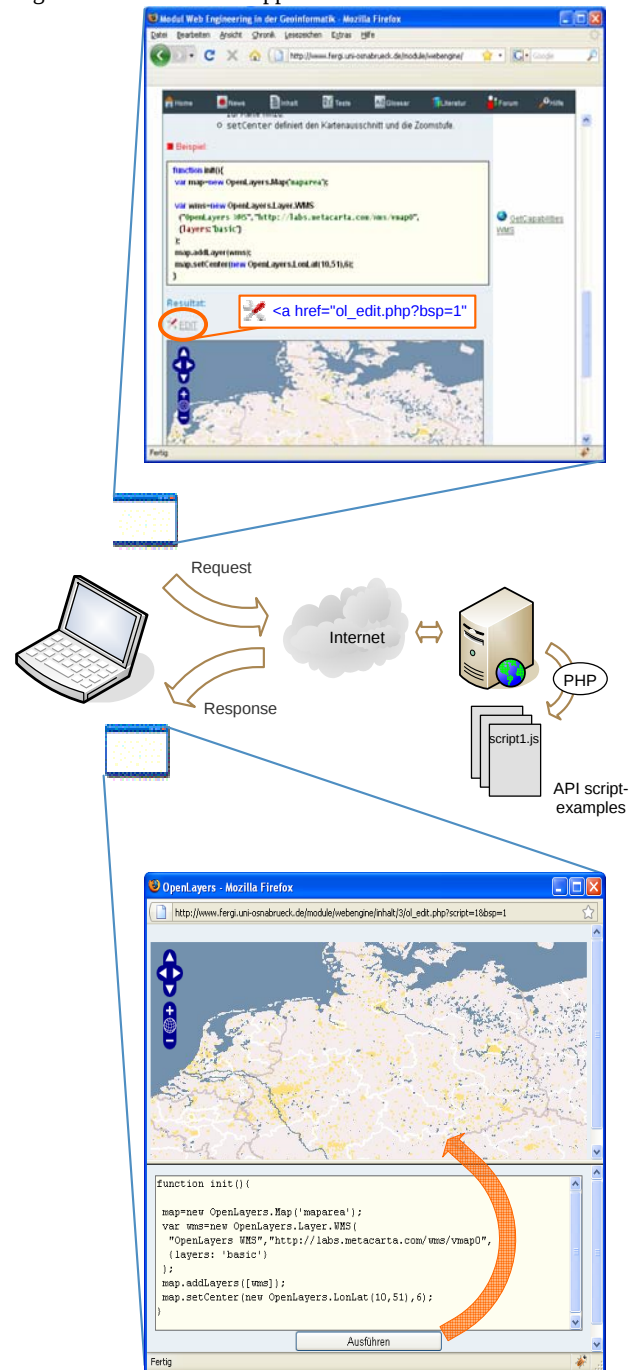


Figure 12. Calling the editor and executing code modifications

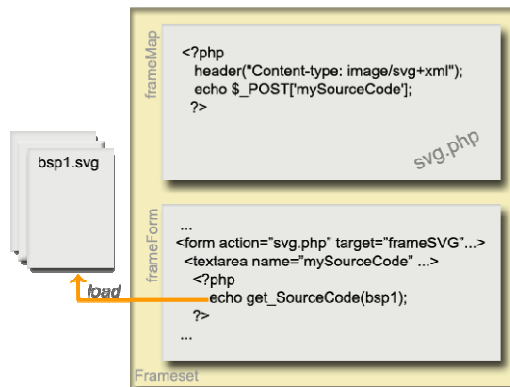


Figure 13. Functionality of the editor in case of SVG

## 5. CONCLUSIONS

Web engineering in the fields of computer science and geoinformatics is nowadays a very broad and fast developing area. Therefore, it is not possible to include all relevant topics into a single e-learning module. Consequently, the presented module “Web Engineering in the Geoinformatics” introduces into a selection of techniques that are of high relevance for web-based map applications. Of further relevance was their suitability for thematic and technical restrictions of the FerGI+ project. As mentioned before, FerGI(+) modules do not reflect the whole GI-curriculum, but concentrate on the design, evaluation and deployment of modules that cover ongoing and applied topics.

Interactivity is in general a very important aspect in online learning environments compared to conventional lectures and books. By interactive animations, the time of learning can be shortened and the comprehension can be increased. Therefore, FerGI(+) modules consist of interactive Flash and SVG animations, Java applets and other comparable elements. For special themes, the development of specific tools can be necessary. For example, in favor of the modules about spatial databases a SQL tool is provided that visualizes stored and retrieved spatial data relations (Krüger & Brinkhoff, 2005). In case of the module “Web Engineering in the Geoinformatics”, the presented editor allows an interactivity that exceeds the potential of normal animations. New skills can be directly exercised and broadened.

An important aspect of the development and provision of e-learning material is to guarantee their sustainability (Krüger et al., 2006). In case of FerGI(+) modules, this goal shall be achieved by a cooperation with the society “Geoinformatics in Northern Germany” (GiN). GiN operates the e-learning platform and the portal to the FerGI(+) e-learning modules. The society is also in charge for marketing, quality and right management. The cooperating universities are responsible for updating the material. In future, the integration of the modules into the curricula of the cooperating and other universities should be intensified. Only a regular use of the modules within the universities will guarantee their regular update.

## REFERENCES

Google Inc., 2009. „Google Maps API Concepts”. <http://code.google.com/intl/en/apis/maps/documentation/> (accessed 15 May 2009)

Grendus, B., 2008. FerGI@KMU: Berufliche, zertifizierte Weiterbildung mit E-Learning-Modulen zur Geoinformatik. *Proceedings of the logOS 2008 „Lernen, Organisation, Gesellschaft - Das eCampus-Symposium der Osnabrücker Hochschulen“*, Osnabrück, Germany, pp. 19-23.

Grendus, B., Schiewe, J., Ehlers, M., 2009. Ein Konzept zur nachhaltigen Sicherung und Pflege von E-Learning-Modulen für die Geoinformatik unter dem Aspekt von Vermarktung und Kooperation. In: Appellath, H.-J., Schulze, L. (eds.): *Auf dem Weg zu exzellentem E-Learning: Vernetzung und Kooperation der Hochschullehre in Niedersachsen*, Waxmann, chap. 17.

Krüger, A., Brinkhoff, T., 2005. Spatial Data Management - Development of e-Learning Modules. *Proceedings 8th AGILE Conference on Geographic Information Science*, Estoril, Portugal, 2005, pp. 207-214.

Krüger, A., Brinkhoff, T., Grendus, B., 2006. Towards a Sustainable e-Learning Solution for GI-Education. In: *Proceedings ISPRS TC VI Symposium “e-Learning and the Next Steps for Education”*, Tokyo, Japan, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, Volume XXXVI, Part 6, 2006, pp. 71-75.

OpenLayers, 2009. “OpenLayers: Free Maps for the Web”. <http://openlayers.org/> (accessed 15 May 2009)

ProgrammableWeb, 2009. “Mashups Tag Search: mapping”. <http://www.programmableweb.com/tag/mapping> (accessed 15 May 2009)

Purvis, M., Sambells, J., Turner, C., 2006. *Google Maps Applications with PHP and Ajax*, apress, 384pp.

Schiewe, J., Hödl, T., Grendus, B., 2006. FerGI: Innovative E-Learning-Materialien zur Aus- und Weiterbildung in Photogrammetrie und Fernerkundung. In: Seyfert, E. (ed.): *Publikation der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V.*, Band 15, Berlin.

Strobl, J., 2004. Erfolgsfaktoren für e-Learning – Lebenslanges Lernen mit Online-Medien. In: Schiewe, J. (ed.): *E-Learning in Geoinformatik und Fernerkundung*, Wichmann Verlag, Heidelberg, pp. 3-10.

Ueberschär N., Winter, A., (2006). *Visualisieren von Geodaten mit SVG im Internet*. Wichmann Verlag, Heidelberg.

W3C, 2003. “Scalable Vector Graphics (SVG) 1.1 Specification”. <http://www.w3.org/TR/SVG11/> (accessed 15 May 2009)

Wilson, T. (ed.), 2008. *OGC KML*. OGC standard 07-147r2, 2008-04-14, 251pp.