

E-LEARNING APPLICATIONS FOR URBAN MODELLING AND OGC STANDARDS USING HTML5 CAPABILITIES

Robert Kaden, Gerhard König, Carsten Malchow, Thomas H. Kolbe

Geodesy and Geoinformation Science, Technische Universität Berlin, Strasse des 17. Juni 135
Sekt. H 12, 10623 Berlin, Germany

{robert.kaden, gerhard.koenig, thomas.kolbe}@tu-berlin.de, carsten.malchow@mailbox.tu-berlin.de

Commission VI, VI/2

KEY WORDS: computer aided teaching, e-learning, OGC standards, HTML5

ABSTRACT

This article reports on the development of HTML5 based web-content related to urban modelling with special focus on GML and CityGML, allowing participants to access it regardless of the device platform. An essential part of the learning modules are short video lectures, supplemented by exercises and tests during the lecture to improve students' individual progress and success. The evaluation of the tests is used to guide students through the course content, depending on individual knowledge. With this approach, we provide learning applications on a wide range of devices, either mobile or desktop, fulfil the needs of just-in-time knowledge, and increase the emphasis on lifelong learning.

1. INTRODUCTION

The use of geodata for navigation systems and visualizations using Google Earth and MS Bing Maps is a daily routine. However, the potential of using spatial information is much higher than only creating 2D/3D visualizations. More complex applications, e.g. the analysis of roof surfaces for photovoltaic suitability, require more than geometric information. 3D geometry together with coherent semantics is the base for virtual 3D urban information models, which decompose our landscape into objects regarding to logical and spatial criteria. They provide an integrative frame for data from different disciplines and application fields, and facilitate a large number of analytic and simulation applications in various disciplines such as city and infrastructure planning, strategic energy and environmental planning, and disaster management.

In this context, standards play an important role in the acquisition and use of urban information models, as the data is typically captured, refined, visualized, and used by different parties and systems. Standards specify the exchange of information from the level of object definition and semantics down to the level of the physical file layout. The use of open standards ensures platform and manufacturer independent management and processing of data. Therefore, the Open Geospatial Consortium (OGC) passes in a consensus process open standards for geospatial content and services that are intended to address the needs of the geospatial community.

Since the specification documents of those standards are normally quite extensive and technical, an increasing demand on learning material including multimedia explanations and examples arises. For this purpose, a variety of platforms exist to present multimedia learning material, such as desktop computers, Macs, Tablet Computers, multimedia MP3 players, PDAs, and mobile phones. The devices run a variety of operating systems such as Windows, Mac OS, Android, iOS, Windows Phone, Symbian, BlackBerry OS, etc. In order to reach a wide audience, developers have to write apps for various platforms, which results in additional costs. HTML5 becomes a consensus over the different devices and operating systems and facilitates multimedia content and interactions with

the user. With HTML5 one can stream audio and video content with no need of third party plugins such as Adobe Flash, which e.g. does not run on Apples mobile devices. It also supports for high quality drawings, charts, and animations, which are useful for providing exercises, quizzes and other learning material.

2. OGC STANDARDS FOR URBAN MODELLING

Despite from the originating discipline, standards for 3D modelling and exchange have to be separated into the two main different classes; 1) (urban) information models and 2) general purpose formats. There is a fundamental difference between both types of standards as the represented objects have completely different semantics, i.e. meanings. While in general purpose formats objects have the meaning of being a container for 0D (point), 1D (line), 2D (surface), or 3D (volume) shaped arbitrary objects with no fixed relation to real world objects or phenomena, in information models objects directly and explicitly correspond to specific real world objects or phenomena.

For urban information modelling purposes, the OGC has released two important standards which facilitate a spatio-semantic representation of cities and landscapes; 1) the *Geography Markup Language* (GML) and 2) the *City Geography Markup Language* (CityGML). CityGML is an international standard for the representation and exchange of semantic 3D city and landscape models and defines a common information model and data exchange format for 3D urban and rural objects. It specifies the classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantic and appearance properties. Included are generalization hierarchies between thematic classes and its aggregations, as well as thematic relations between objects. This thematic information goes beyond virtual reality or computer graphic standards and facilitates sophisticated analysis tasks in different application domains. CityGML is implemented as an application schema of the Geography Markup Language 3.1.1 (GML3; see [Cox et al., 2004]), the extensible international standard for geodata exchange and encoding issued by the OGC and the ISO TC211.

GML facilitates the representation of objects including geometry, attributes and relations. GML and CityGML are applications of the *eXtensible Markup Language* (XML) and bases on a number of further standards from the ISO 191xx family, the OGC, the W3C Consortium, the Web 3D Consortium, and OASIS [Gröger et al., 2008].

The OGC GML and CityGML standards are technical documents for software developers to build open interfaces and encodings into their products. To enhance the usage of standardised virtual 3D city models for multiple purposes, the Institute for Geodesy and Geoinformation Science at Technische Universität Berlin is preparing multimedia learning material for the relevant standards including lectures and exercises.

3. PREVIOUS E-LEARNING IMPLEMENTATIONS

The rapid technological development of the computer industry and the associated new methods for content design especially allow the distribution of multimedia enhanced content via the Internet. Whether as a pure online course or in combination with a presence seminar or a smart phone, learning today can be easily integrated into everyday life. Today, technologies creating web-based material make heavily use of Adobe Flash or video podcasts. But there is a clear trend to avoid browser plugins for adding animation, video, and interactivity in near future. In this section, we present and discuss common web techniques which we have used to provide learning material about urban information modeling using OGC standards.

3.1 Adobe Flash

Adobe Flash is an advanced authoring tool for creating interactive and animated multimedia content. In the educational field, Flash is widely used, because it is suited for the visualization and interrelation of processes. It is based on animated vector graphics and therefore has low memory demands, requiring relatively low transfer rates. A further advantage is related to the good scalability of display sizes without loss of quality. The central work space in Flash, the stage, facilitates arranging media objects (text, graphics, sound, video, etc.) along the timeline. Their behavior is controlled with the help of visual programming tools, e.g. for tweening.

Flash has its own scripting language called ActionScript, which allows to interactively control the sequence. Beyond the possibility of film timing, ActionScript offers several other functions, e.g. for dynamic loading of external files, complex calculations, communicating with a server and much more. For interactive learning content, user interactions like multiple choice questions can be easily integrated into the web application. Flash projects are exported to the propriety binary file format SWF. To replay the embedded content within a web page, specific plugin software for Internet browsers is needed. The Adobe Flash Player is one of the most widely used browser plugins. It is available for the 32 bit and 64 bit versions of almost all currently popular desktop and mobile operating systems – except for Apples iOS.

Driven by the increasing demand on learning material about urban information modelling using GML and CityGML, a desktop e-learning course was developed at the TU Berlin in 2008. The initiative to setup the course was pushed by the EuroSDR committee who decided to offer online training during the sixth and seventh round of their annual courses. The

course consists of lectures on-demand which were developed as Flash animations including animated slides and synchronised videos showing the lecturer. Therefore, the software iSpring Ultra, a PowerPoint plugin for exporting animated content as Flash, has been used [König and Kolbe, 2008].

Providing learning material with Adobe Flash has a lot of advantages but also disadvantages. Indeed, Adobe Flash is still very much alive and well suited for e-learning and there is nothing at this time that can provide similar rich interactivity in connection with dynamic content like slide show presentations, movies, audio streams, etc. The scope of HTML5 is to provide similar functionalities; however, this is not fully the case yet due to the lack of adequate development environments and browser support. Nevertheless, Adobe Flash has also some significant disadvantages. Playing Adobe Flash files requires a browser plugin and is not supported by Apples mobile operating system iOS. Since Apples mobile devices became very popular in society over the whole world, the development and spreading of Adobe Flash web content has been dampened as it cannot be viewed with these devices. Moreover, the developed Flash lectures on-demand require a permanent Internet connection and were confined to a desktop computer.

3.2 Video Podcasts

Podcasting is an excellent medium to bring learning material to the users' devices, anytime and anywhere. The podcast technology facilitates lectures on-demand including animated slides and audio streams capable for desktop computers and current mobile devices of all brands. Podcasts are an excellent medium for mobile learning, to bring lectures to the daily train ride which one might go or anywhere else. Beside a free podcatcher program, e.g. iTunes or any browser for desktop or mobile devices, podcasts do not need any specific software or licence and are independent from permanent Internet connection. Podcasts need to be downloaded once and can be played offline anywhere and any often. Not only the consumption of podcasts, but also the creation is quite easy.

Due to the disadvantages of Adobe Flash, a new procedure was developed to create podcasts instead of the Flash-based lectures on-demand. A podcast for animated lectures requires three elements; 1) a media file which is in that case a video, 2) a RSS feed to describe the podcast and its location in the web, and 3) a publication platform.

The video file was created using *Adobe Captivate 4* which is explained in detail in König et al. [2010]. The video bases on animated PowerPoint slides which are synchronised with an audio track including explanations. We decided to use a synthetic voice instead of video or audio taping a human being, in order to be more flexible with updating the content. The audio track for each slide was created using the Text-to-Speech functionality in Adobe Captivate 4, whereas the scripts including the text was derived from the original spoken video tapes taken from the old Flash lectures on-demand. For the transcription, the Speech-to-Text software *Dragon Natural Speaking* was used with moderate success. Our experiences have shown that it is necessary to check and correct the text files by parallel listening to the original video files. This phase was the most time consuming part with approximately 2/3 of the time. However, using a transcript from a real presentation makes the explanations livelier than just verbalizing a text in written language. The PowerPoint animations were then synchronized with the generated synthetic audio tracks using Adobe Captivate 4. Unfortunately, Adobe Captivate 4 only

facilitates the video export as an AVI so that we have used the Free MP4 Video Converter to render a MP4 video, which led to some loss of quality. Figure 1 gives an overview on the processing chain. Finally, we created an RSS feed, which is an XML structured text file containing metadata like the authorship, a short description of the content, the date of publishing, the web address (URL), the file size, and the file type of the media content. The RSS feed has been published on our web page (<http://www.igg.tu-berlin.de/podcasts/rss.xml>) and with iTunes in order to reach a broader community in our profession [Kaden and König, 2011].

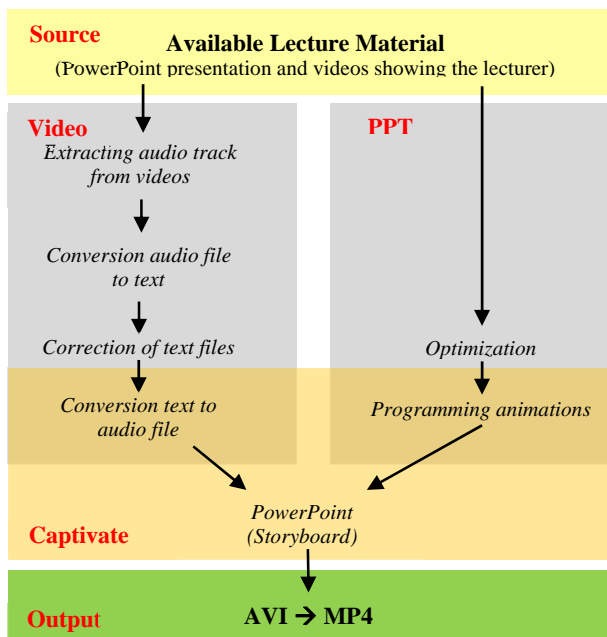


Figure 1: Flowchart for podcast production

Video podcasts overcome the drawbacks of Flash and thus are very suitable to provide learning material which is running on all popular systems, from desktop computers to mobile phones, with all current operating systems, independent from a permanent internet connection. Since one has subscribed the RSS feed, the user will get automatically updated files and new content, which is a big step towards sustainability. However, podcasts only facilitates providing text, audio, or video content without any interactive functionalities supporting exercises like multiple choice tests.

4. HTML5 CAPABILITIES AND IMPLEMENTATIONS FOR E-LEARNING

After Adobe no longer pursues the Flash platform for mobile browsers and prefers HTML5, there is a need for a closer look at the HTML5 specifications. At the moment HTML5 still is in the process of standardisation. According to the schedule of the W3C HTML5 will become officially adopted in 2014, that is, will be approved as W3C Recommendation. The steps which a technical report has to pass, starting from the initial Working Draft via Last Call Working Draft, Candidate Recommendation, and Proposed Recommendation to finally become a W3C Recommendation, are long. At first glance it seems that HTML5 is in Alpha mode, since only two modules have passed as recommendation and about the half is available as Working Draft. But the other half is available as Candidate or Last Call. That means that HTML5 has in fact already adopted a final

state, which is comparable to a release candidate. In most web browsers, HTML5 is implemented by now – albeit incomplete.

4.1 HTML5 Capabilities

The HTML5 technology introduces several new elements which allow to better structure complex websites and adds helpful functions, e.g. for embedding multimedia more comfortable without using plugins. It includes established web technologies such as the Mathematical Markup Language (MathML) for integrating mathematical formulae or Scalable Vector Graphics (SVG) for placing static, interactive, and animated two-dimensional vector objects. The HTML5 specification includes attributes and events for drag and drop and the canvas element allows for dynamic, scriptable rendering of 2D shapes and bitmap images.

The question whether one can already use HTML5 or not, stands or falls with the support of the technologies by the web browser. Table 1 shows how well the most used browsers for desktop and mobile devices already support HTML5. The information is adopted from the website <http://html5test.com>, which on access automatically displays your browser score. The score is calculated by testing for the many new features of HTML5 – at present maximum score is 500. Bonus points are given for supporting audio and video codecs, which are not part of HTML5 specification. Also support of SVG and MathML is rated as bonus. The HTML5 test website is maintained by Niels Leenheer, a Dutch software developer, who updates it regularly.

OS		Score	Bonus
Chrome 18.0		400	13
Firefox 11.1		345	9
Opera 11.60		338	9
Safari 5.1		317	8
Internet Explorer 9		138	5
Opera Mobile 12.00	Multiple platforms	369	11
Firefox Mobile 10	Multiple platforms	325	9
iOS 5.1	iPhone, iPad, iPod Touch	324	9
Android 4.0	Samsung Galaxy Nexus	273	3
BlackBerry OS 7	Bold 9900, ...	273	3
Windows Phone 7.5	Samsung Omnia W, LG E906, ...	138	5

Table 1: Browser ranking related to HTML5 support derived from <http://html5test.com> (15 April 2012)

Main new features relevant for designing e-learning material are described briefly in the following sections

4.1.1 Audio / Video

Up to now, videos that have been presented on websites mainly using Flash movies or videos requiring plugins. With HTML5 videos are supported by the browsers natively, a plugin is no longer needed. `<audio>` and `<video>` elements are introduced for integration. However, no video codec and container format have been specified yet, which led to controversial discussions. Since the Internet should only be based on open technologies that are free of charge the codec question became a problem. MPEG LA holds the powerful H.264 patent (with Apple, HP, Microsoft and further companies as licensors). The less powerful, but free ogg/Theora codec used by Firefox and Opera (but also supported by Google Chrome) loses its influence, because with VP8/WebM a new competitor came into the

game. The new format (sponsored by Google) was designed to provide royalty-free, open video compression and now is accepted by a variety of companies – except Apple and Microsoft. Meanwhile a plugin for Internet Explorer is available and a plugin for Safari is announced from the WebM project. In response MPEG LA gave notice “that its AVC Patent Portfolio License will continue not to charge royalties for Internet Video that is free to end users (known as Internet Broadcast AVC Video) during the next License term from January 1, 2011 to December 31, 2015” (<http://www.mpegla.com/Lists/MPEG%20LA%20News%20List/Attachments/226/n-10-02-02.pdf>). Table 2 gives an overview on the video codices supported by the mostly used browsers for desktop and mobile devices.

OS	ogg	H.264	VP8
IE 9.0	no	yes	no
Firefox 11.1	yes	no	yes
Safari 5.1.5	no	yes	no
Chrome 18.0	yes	no	yes
Opera 11.62	yes	no	yes
iOS	no	yes	no
Android	yes	yes	yes
IE Mobile	no	yes	no

Table 2: Native video codec support by browser engines

4.1.2 Mathematical Markup Language (MathML)

MathML is an application of XML for describing mathematical notations and capturing both its structure and content. It was released as a W3C Recommendation in 2010. For integration of mathematical formulae HTML5 tags `$...$` have to be used that embed MathML presentation elements. Following code lines are necessary to obtain the simple equation:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>MathML embedded in an HTML5 file</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <p>
      <math>
        <mrow>
          <mi>x</mi> <mo>=</mo>
          <mfrac>
            <mrow>
              <msqrt>
                <msup>
                  <mi>b</mi> <mn>3</mn>
                </msup>
              </msqrt>
            </mrow>
            <mn>2</mn>
          </mfrac>
          <mi>a</mi>
        </mrow>
      </math>
    </p>
  </body>
</html>
```

$$x = \frac{\sqrt{b^3}}{2a}$$

`<mi>...</mi>` is used for identifiers, `<mo>...</mo>` for operators and `<mn>...</mn>` for numbers. Layout elements are used for rows `<mrow>...</mrow>`, fractions `<mfrac>...</mfrac>`, roots `<msqrt>...</msqrt>`. For marking up a base expression and its superscript `<msup>...</msup>` is necessary. The character entity `⁢` used in the example indicates that there are special spacing rules between the 4 and the a, and that the 4 and the a should not be broken onto separate lines. The majority of desktop and mobile browsers now support embedding MathML in HTML5 documents.

4.1.3 Scalable Vector Graphics (SVG)

SVG 1.0 standard for 2D vector graphics became a W3C Recommendation in 2001. Although it's an old standard, web browsers so far treated it only half-heartedly. HTML5 gives acceptance of SVG a push since it is part of the HTML5 specification. At present SVG is redesigned integrating new features such as filters, gradients, clipping, and other effects. The work is about to be finished, SVG 2.0 is scheduled for recommendation next year (<http://www.w3.org/Graphics/SVG/WG/wiki/Roadmap>). The following code is used to display a green rectangle:

```
<!DOCTYPE html>
<html>
  <head>
    <title>SVG</title>
  </head>
  <body>
    <svg width="250" height="100">
      <rect
        width="250" height="100"
        fill="green"
        stroke="black" stroke-width="4"
      />
    </svg>
  </body>
</html>
```

The specification of SVG 1.1 (Second Edition – Recommendation 16 August 2011) published by the World Wide Web Consortium is available at <http://www.w3.org/TR/SVG11/>. Now most web browser for desktop as well as mobile applications support and render SVG markup natively.

4.1.4 Canvas

One of the key features of the new HTML5 standard is the *canvas* element that allows for dynamic, scriptable rendering of 2D shapes and bitmap images. A canvas 2D element will allow performing drawing operations using JavaScript in HTML5 on a defined area. It is embedded by the `<canvas>` tag that has to be named with an *id* followed by the attributes *width* and *height*. In addition, a JavaScript function is needed that includes the sequence of methods necessary for visualisation. A green 2D rectangle is represented by the following code:

```
<!DOCTYPE HTML>
<html>
  <head>
    <script type="text/javascript">
      window.onload = function() {
        var canvas = document.getElementById("myCanvas");
        var context = canvas.getContext("2d");
        context.fillStyle = 'rgb(0, 255, 0)';
        context.fillRect(50, 70, 280, 200);
      };
    </script>
  </head>
  <body>
    <canvas id="myCanvas" width="578" height="200">
      <strong>Browser doesn't support canvas! </strong>
    </canvas>
  </body>
</html>
```

To get an overview about available canvas features, one may have a closer look at the HTML5 Canvas Cheat Sheet (<http://www.canvasdemos.com/2009/02/24/canvas-cheat-sheet>).

There are already initial implementations, drawing on the canvas element and animations to present. It can draw lines, paths, text and images are drawn on the drawing surface. The implementation of WebGL, a JavaScript API for rendering interactive 3D graphics also uses the HTML5 *canvas* element [Khronos, 2011]. All the major browser vendors implemented the Canvas 2D element for desktop as well as for mobile applications.

4.1.5 Further HTML5 features

Of course HTML5 offers a variety of further new features, such as offline web applications that are available in case the Internet connection breaks down, native drag and drop, document editing, new form types, geolocation, and more. You will find all features described detailed in the W3C documentation [World Wide Web Consortium, 2012]. To get an example based overview of main features visit the website of the Google project *HTML5 Rocks* (<http://www.html5rock.com>). In an interactive HTML5 based slideshow (<http://slides.html5rocks.com>), one can make own experiences and study the source code for getting knowledge how things work.

4.1.6 Support for CSS3

Cascading Style Sheets (CSS) for web design are a W3C Recommendation since 1996. The main advantages – the separation of context from styling, site-wide consistency, minimising code, easy reformatting, etc. – boost CSS to become generally accepted. Nowadays professional websites are hardly imaginable without taking advantage of CSS and all web browsers support it. In the meantime CSS Level 3, based on CSS2.1 specification [World Wide Web Consortium, 2011a] is in the standardisation phase. In order to make the process manageable it has been divided into several modules. The list of modules and their current status can be viewed on the website <http://www.w3.org/Style/CSS/current-work>.

Useful changes compared to CSS2.1 are new powerful selectors, multi-column layouts, transforms allowing scaling, rotating, and positioning any object (usually images or texts), transitions and animations (colour fading effects), linking font files to display text, media queries for customising styles also on characteristics of the user's display (e.g. viewport width), and several new visual effects (semi-transparent backgrounds, drop shadows, etc.)

4.2 E-Learning Implementations using HTML5

With the ongoing establishment of HTML5, we decided to combine our e-learning implementations with the new HTML5 capabilities in order to provide an integrated platform that supports animated lectures on-demand in combination with interactive exercises and quizzes. HTML5 seems to become a consensus over all kind of devices and operating systems from competing manufacturers and we believe that HTML5 provide the largest potential, not only in technical aspects but also in aspects of interoperability and sustainability.

4.2.1 Creating Video Lectures for HTML5

Since we have started preparing learning material for the implementation with HTML5, we have decided to re-create the MP4 videos which we have developed for the podcasts. The reason was that some videos were too long. From the didactic point of view one video lecture should not be longer than 10 to 15 minutes, because on average the attention span to pick up new information is limited due to fast fatigue of perception [Süß et al., 2002]. Therefore, the video length of the new video lectures on-demand is restricted to 5 to 10 minutes. However, the re-creation of the new shorter videos was straightforward since we use Adobe Captivate for the creation of the video podcasts. The Adobe Captivate project includes all slides which are synchronized with separate synthetic voice audio tracks. This allows to pick single animated slides with their audio tracks and to render arbitrary videos with adjusted speech at the beginning and the end of the new sections.

With the re-creation of the video files for the lectures on-demand, we have updated to Adobe Captivate version 5.5 since it enables a direct export of MP4 videos. The Adobe Captivate projects which have been developed in version 4 could be read without problems. The used H.264 video codec is at most compatible for all tested platforms [König et al., 2010]. However, though the HTML5 video-tag is supported by the five common web browsers Internet Explorer 9, Firefox 11, Chrome 6, Safari 5, and Opera 11, not all video codices are implemented. MP4 videos are not supported by Firefox and Opera but by Internet Explorer, Google Chrome and Safari (see chapter 4.1.1). Therefore, the MP4 video has to be converted to the ogg-Format in order to be able to run with Firefox and Opera, too. That means, at least two copies of the same video file have to be available.

In average, one minute video has a size of approximately 3 MB what leads to a total size of 15 MB to 30 MB for one lecture. Assuming an Internet connection DSL 6000 with about 6 Mbit/s, the download time would not be longer than one minute.

4.2.2 Integrating Learning Material with HTML5

For the implementation of the video lectures, Adobe Labs offers an experimental *HTML5 converter for Captivate 5.5* (preview 2) that transfers SWFs into HTML5 format. It promises an automatic creation of HTML5 content from Adobe Captivate projects. Therefore, the lecture sections have to be exported as a SWF Flash file which is to be converted to HTML5 afterwards. Unfortunately, the tested beta version has not led to suitable results. We tested different settings, amongst others, settings which should be optimized for Apples iPad, but the video lecture did not run on our test devices. Thus, we implemented the video lectures using the new HTML5 video-tag element manually. Since the incompatibility of some video formats with web browsers, two different video sources were implemented as follows.

```
<video id="video" width="600" height="400"
      controls="controls">
  <source src="P10SolidSum.ogg" type="video/ogg" />
  <source src="P10SolidSum.mp4" type="video/mp4" />
</video>
```

The prepared video lectures are now built into our new e-learning web page based on pure HTML5 including CSS3 and JavaScript but without any proprietary plugins for videos, animations and interactions. This web page consists of separate lectures of about 90 Minutes. Each lecture contains of a HTML5 presentation including single slides. The slides are structured as HTML5 div-layers, designed with CSS, and are controlled by navigation buttons for forward and backward control, encoded with JavaScript. Each slide contains either a short video lecture, further written explanations, or an exercise regarding to the topic of the video lectures.

The short exercises are implemented using HTML5 forms for the input of answers and JavaScript for processing and checking the result. These exercises are short and pregnant and repeat the content of the video lecture. Following types of exercises are integrated into our training course:

- Short answer (input required by keywords)
- Cloze (supplement required by spaces)
- Multiple Choice (choose one answer from several alternatives)
- Matching (Drag & Drop exercise)

Learning success for each lesson is controlled by one of the different type mentioned before based on HTML5 code (see figure 3). You will find

Part 3 - Appropriate Feature Classes for given Objects

Within the final task of Exercise III, you should get familiar with the "bible" of CityGML, the CityGML Encoding Standard document version 1.0.0. Please get familiar with the thematic model and feature classes in order to find a solution for the assignment.

CityGML Encoding Standard document version 1.0.0

Assignment:
Please answer the short question to skip to the end.

For your project, you intend to model the following objects in LoD 4. Which CityGML feature classes would you choose? Match the listed objects with the appropriate CityGML feature classes.

Roof: Choose...	Forest: Choose...
Chimney (Single Family House): Choose...	Grass Strip between two Lanes (Road): Choose...
Chair: Choose...	Industrial Chimney: Choose...
Traffic Light: Choose...	Coffee Machine: Choose...

Please match the above pairs

Figure 3: Exercise example

5. CONCLUSIONS AND OUTLOOK

Up to now nearly all authoring and development tool for e-learning use Flash formats as output, but some disadvantages cannot be neglected. HTML5 is a promising technology with increasingly dynamic growth and acceptance. Up to now usage is some kind of surprise and challenge, dependent on how HTML5-friendly the web browsers are. But features are more and more supported, so it's a question of time since they are fully compliant with the – hopefully soon W3C recommended – standard. If the development of teaching material is restricted to desktop-based modules then the traditional use of Flash files could be an alternative, but focussing also on mobile learning too, that gains a significant upturn, there is no alternative to HTML5 usage. For the development of nearly platform independent learning material HTML5 code cannot be waived.

Today HTML5 is available on every desktop-based browser but also almost all mobile devices on the market have browsers that support a large part of HTML5. It's a good choice especially for mobile devices, because rendering HTML5 code is generally not very processor-intensive compared to alternatives we have seen so far. Of course this pushes the acceptance of mobile device manufacturers to use HTML5 since they are oriented to the demand of users who expect increased speed and multitasking. It is therefore sensible to use technologies that claim to represent content that waste less bandwidth and processing power. The short product cycles of mobile devices usually entails the adoption of an updated browser and encoder in order to take advantage of the new hardware features, and HTML5 will benefit from this.

What is still missing, however, are powerful tools that support designers and developers in the creation of HTML5 based learning materials. Even with the market leader Adobe here is greater need for action. In summary the definition of HTML5 as the basis for the creation of websites is a step in the right direction and will have more influence in near future.

6. REFERENCES

- Cox, S, Daisy, P, Lake, R, Portele, C, Whiteside, A (2004) OpenGIS Geography Markup Language (GML3.1), Implementation Specification Version 3.1.0, Recommendation Paper, OGC Doc. No. 03-105r1
- Gillenwater, Z.M. 2011: Stunning CSS3: A project-based guide to the latest in CSS, New Riders Press. 320 pp.
- Gröger, G., Kolbe, T. H. ; Czerwinski, A., Nagel, C. 2008: OpenGISR City Geography Markup Language (CityGML) Encoding Standard / Open Geospatial Consortium Inc. [http://portal.opengeospatial.org/files/?artifact_id=28802], International OGC Standard Version 1.0.0
- Kaden, R.; König, G. 2011: Sustainable Development of E-Learning Material for OpenGIS Standards. *Int. Arch. of the Photogrammetry, Remote Sensing and Spatial Information Science*, 38-6/W27. 6 pages [<http://www.isprs.org/proceedings/XXXVIII/6-W27/>].
- Khronos Group, 2011: WebGL Specification, Version 1.0, 10 February 2011, [<https://www.khronos.org/registry/webgl/specs/1.0/>].
- König, G.; Kolbe, T. H. 2008: Using Standards and WEB 2.0 Technologies for E-Learning Modules – Joint Development of an EuroSDR Course on CityGML. *Int. Arch. of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol.XXXVII-Part B6a, pp.241-246, 2008. [http://www.isprs.org/proceedings/XXXVII/congress/6a_pdf/7_ThS-16/05.pdf]
- König, G.; Kaden, R.; Kolbe, T. H. 2010: E-Learning Everywhere with Podcasts and Co. – The CityGML Training Course for Mobile Devices. *Int. Arch. of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol.XXXVIII-Part 6, 2010. [http://www.isprs.org/proceedings/XXXVIII/part6/papers/Koenig/2010-Enschede_Koenig-Kaden_paper.pdf]
- Mavrody, S. 2012: Sergey's HTML5 & CSS3: Quick Reference. HTML5, CSS3 and APIs. (2nd Edition), Belisso. 218 pp.
- Süß, C., Kammerl, R., Freitag, B., Weitl, F., 2002: Presenting Complex e-Learning Content on the Web: A Didactical Reference Model. In M. Driscoll & T. Reeves (Eds.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2002* (pp. 1018-1025).
- World Wide Web Consortium 2010: Mathematical Markup Language (MathML) Version 3.0, W3C Recommendation 21 October 2010, [<http://www.w3.org/TR/MathML/>].
- World Wide Web Consortium 2011a: Cascading Style Sheets Level 2 Revision1 (CSS 2.1) Specification, W3C Recommendation 7 June 2011, [<http://www.w3.org/TR/SVG11/>].
- World Wide Web Consortium 2011b: Scalable Vector Graphics (SVG) 1.1 (Second Edition), W3C Recommendation 16 August 2011, [<http://www.w3.org/TR/SVG11/>].
- World Wide Web Consortium 2012: HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Working Draft 29 March 2012, [<http://www.w3.org/TR/html5/>].

All links cited in the paper are accessed on April 15, 2012