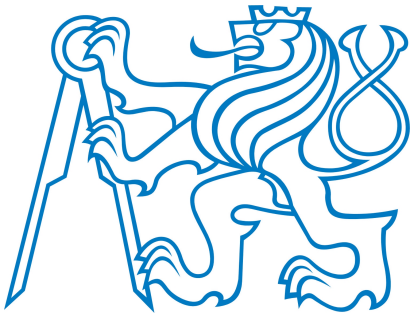
3D Web Technologies And Their Usability for The Project 3D Mobile Internet

Technical Report

Beata Turoňová

Research and Development Center for Mobile Applications
Faculty of Electrical Engineering
Czech Technical University in Prague

December 2009



Nadace
Vodafone
Česká republika

Contents

Abstract

This report presents a very brief overview of few existing 3D technologies, which enable to present 3D data on the internet, and consider their usability for the project 3D Mobile Internet. First, it introduces some of the 3D technologies and their key features in general. These technologies, their advantages and disadvantages, are then again reviewed in terms of the project. Finally, based on project's desired properties, the most suitable 3D technology is recommended

Acknowledgement

The project is kindly supported by Vodafone Foundation Czech Republic.

1 Introduction to 3D Web Technologies

Both hardware and software support for displaying realistic 3D graphics is growing every day. Therefore, there exist many technologies, which enable to present 3D data on the internet. In this first part, only some of them are briefly introduced. There is a short description to each technology with their key features, advantages and disadvantages.
Since the main focus is on 3D graphics, technologies oriented mainly on development of interactive web applications (e.g. Adobe Flash [25]) are left out.

1.1 VRML

VRML (Virtual Reality Modeling Language) [1] is one of the oldest graphic formats for representing 3D scenes mostly on the internet. Its first version VRML 1.0 was released in 1995 and was based on the library OpenInventor [15]. The second and last version was released in 1997 and it is called VRML97 (or VRML 2.0). In the same year was VRML97 also accepted as international standard by the ISO. VRML was and still is supported by web browsers and most of the VRML viewers are for free. It is a multiplatform technology and is supported by 3D modeling programs such as 3ds Max [23] and Maya [22]. Despite this, VRML never became as popular as expected. While realtime 3D graphics kept improving over the time, the VRML properties remained from most part the same and therefore not sufficient. Nowadays it is considered old and attention is paid mainly to its successor X3D [2].
VRML is based on declarative programming language. The 3D scenes, so called "worlds", are represented in a text file with an extension wrl (or wrz for compressed file). Worlds are structured using specific nodes. The whole structure is similar to a tree structure. Since children inherit properties of their parents, we can observe also properties of objective oriented programming. Dynamic or static world is therefore composed from 3D objects. These objects could be both passive and active, which can be combined e.g. with images, video and sound. Objects could also be used as links to another world.
Except description of the world, the language also specifies the way user can move (walking, flying, exploring), it supports automatic navigation through the virtual world (using so called viewpoints) and also describes reaction on events triggered mostly by the user. VRML supports scripting as well - most used is Java or JavaScript (ECMAScript). This allows adding desired properties to the world.

- Advantages
  - multiplatform

  - is accepted as international standard
  - was designed especially for representing 3D content on the internet
  - is supported by 3D modeling programs such as 3ds Max and Maya
- Disadvantages
  - has no real integration with HTML
  - its last version is from 1997 and has not been improved since then

1.2 X3D

X3D (Extensible 3D) [2] is a successor of VRML. It is ISO standard file format for representing 3D graphics using XML syntax. The 3D scenes could be also represented using VRML syntax or binary form (for use in application). The latest release is from July 2008.
X3D improves upon VRML with new features, advanced APIs and componentized architecture using profiles. The profiles are defined to support modularity of X3D architecture. Developer can decide which profile suits him better and use it. The basic profile is interchange profile for communication between applications. It does not contain run time model for rendering and therefore is really easy to integrate to any application. The interactive profile provides basic interaction with 3D scene by using e.g. sensors or timing. The immersive profile enables full 3D graphics and interaction, including audio support and scripting (C, C++ and Java). The full profile includes all defined nodes (also extensions to VRML such as NURBS, GeoSpatial, Humanoid Animation). There also other profiles and new can be created and standardized.
X3D supports also 2D graphics, CAD data, video files, animation, navigation, user-defined objects, physical simulation. It has been officially incorporated within the MPEG-4 multimedia standard. Though it is standard for representing 3D data on the web and it was released few years ago, it has never become as popular as expected (similar to VRML). The development is not so flexible and quick and it seems it has troubles to keep up with new technologies. There are many X3D groups trying to create new standards for specific fields such as medicine, geography and others, but their work seems to be not as consistent as it should be.

- Advantages
  - multiplatform
  - it is standard for representing 3D content on the web (and therefore is also very well specified)

  - it is divided to profiles
  - it is supported by many 3D modeling program such as Maya or 3ds Max

- Disadvantages
  - still not so popular among web users
  - development seems to be not enough flexible

1.3 3DMLW

3DMLW (3D Markup Language for Web) [3] is a file format based on XML developed by 3D Technologies R&D [12]. It is designed for creating and representing both 3D and 2D interactive content on the internet. The interactive scene is represented by a text file with XML syntax with an extension .3dmlw.
3DMLW is not only the markup language it is whole technology containing also scripting support for dynamic and interactive content, style sheets for designing, plug-in for web browsers, editor for creating .3dlmw documents and model viewer (for .3ds, .blend, .an8, .obj files). This technology is open source licensed under GPL.
Except plain or formatted text and 3D objects, 3DMLW allow publishing also pictures (.jpg, .png, .tga), audio files (.wav, .ogg) and interactive content through scripting (scripting language Lua). In the future also video support is planned. For rendering OpenGL [26] is currently used, but support for DirectX [27] is also planned. The 3DMLW technology is quite new - first stable release was in June 2009. This is both good and bad. The good thing is that this technology is up to date and still developing. The bad part is, that thought promised, there is no concrete time plan for intended features.

- Advantages
  - combines 3D and 2D content
  - the similarity to XHTML makes it easy to understand
  - it is up to date with new technologies

- Disadvantages
  - currently only Windows platform is supported
  - currently no video support
  - relatively new - many bugs and not so common among users

1.4 XML3D

XML3D [4] is a new file format for representing interactive 3D content on the web. It is being developed by Computer Graphics Group of the Saarland University [13] in Germany. From its name it is obvious, it was extended from XML. The first release is due in January or February 2010.
Since there has not been any release yet, the specification is not available. However, there were few presentations of this technology with very promising examples. XML3D should be multiplatform and should be direct part of web browsers. Right now, it is designed as plug-in for web browsers, but in the future it is supposed to be directly in web browsers (agreement with Mozilla [17] has already been done, discussion with others is in progress). It is also expanding to WebKit [16]. This makes XML3D very convenient for the users, because no additional downloading or installing of plug-ins will be necessary.
The 3D scene is represented by XML syntax and it is direct part of HTML code. There is support for simple objects like spheres, cubes, cylinders etc. and for representation with vertices and triangles. The import of whole 3D models from other programs such Maya is also possible. The syntax is very intuitive for everyone, who has ever worked with any 3D scene representation.
There is of course support of audio files, video files and animation. Actually, full integration of realtime digital media should be in XML3D. It should be multimodal (3D audio, nimation, AI, physics...). XML3D uses realtime ray tracing for rendering. which brings another advantages as portable appearance or unified shading tehcnology.
Similar to 3DMLW, XML3D combines 2D and 3D content, which could be fully interactive. Objects in scenes could be used as links to other pages and links on the web pages could be used to change the content of 3D scene.
The big disadvantage of this technology is the fact that it is bright new and the first release could contain many bugs or could miss some of the promised features. On the other hand, the discussion with W3C and Web3D about standardization of this format is in progress and so far it looks very well for XML3D.

- Advantages
  - it combines 2D and 3D content
  - it should be direct part of web browsers
  - it is multiplatform and multi-modal
  - it is up to date
  - it uses realtime ray tracing for rendering
  - 3D scene is put to HTML DOM

- Disadvantages
  - it is new - many practical problems could show up after first release

1.5 WebGL

WebGL [5] is described as OpenGL ES 2.0 [14] for the web. It is cross-platform, web standard for a low-level 3D graphics API. As mentioned above, it is based on OpenGL ES 2.0 and the accelerated 3D content is contained in Canvas element of HTML5. WebGL is a new technology - first working draft was released in December 2009. The developer is WebGL Working Group [10], which includes Apple [28], Google [19], Mozilla [17] and Opera [18].
Unlike previous methods, the 3D content is in a form of application - it is easy to use for developers familiar with OpenGL ES, but it is little bit too complicated for common users (the previous technologies could be used for creating static scenes by 3D graphics fans end web designers without any knowledge of programming). The advantage of this technology is that it is plug-in free. It is direct part of web browsers, including also for us interesting WebKit [16]. It takes to consideration also displaying 3D content on mobile phones. There is a tight integration with HTML content, including layered compositing, interaction with other HTML elements, and use of the standard HTML event handling mechanisms. There is no information about multimedia support outside the HTML5 properties.

- Advantages
  - directly in web browsers
  - up to date
  - multiplatform
- Disadvantages
  - new - contains many bugs
  - 3D content is in form of an application
  - 2D and 3D content is separated
  - so far quite negative responses among users

1.6 O3D

O3D [6] is an open-source JavaScript API for creating interactive 3D graphics applications that run in a browser window - games, ads, 3D model viewers, product demos, virtual worlds. It was created by Google and is open source licensed under BSD.
The 3D content of the O3D application could be created directly in the code (using classic representation with vertices, triangles, normals, shaders etc.) or imported from 3D models created in Maya [22], 3ds Max [23], Google SketchUp [24]. However, the imported files have to be converted in COLLADA files first. The 3D content is in a form of application - you have to always write a program to display even a simple non-interactive 3D object (e.g. cube). For rendering either OpenGL or Direct3D is used.
The O3D JavaScript application code is completely contained in an HTML document that is loaded into a web browser. However, the communication between 2D web content and 3D application is not that simple as e.g by 3DMLW or XML3D. In other words, O3D is more about running whole 3D applications on the web rather than presentation of the 3D scenes as a natural part of web content.
Since moreover JavaScript syntax is used for displaying the 3D content, there is no real standardization.

- Advantages
  - multiplatform
- Disadvantages
  - so far no standardization
  - 3D content is in a form of application

1.7 U3D

U3D (Universal 3D) [7] is a compressed file format standard for 3D computer graphics data. It was developed by 3D Industry Forum, which is a group of companies and organizations including e.g. Intel [29], Adobe Systems [32], Right Hemisphere [30] and Boeing [31]. Initial release was in August 2005 and latest release is represented by 4th edition from June 2007. The format was extended from XML.
From developer's name one can easily conclude, that the main focus by developing was on usability of 3D graphics in industrial applications. U3D therefore could be seen as standard for sharing 3D data among many applications in the industry. Since this easy exchange of 3D format was the

main goal of U3D, there is no real support for representing 3D content using U3D on the web.

1.8 COLLADA

COLLADA (Collaborative Design Activity) [8] was designed to establish an interchange file format for interactive 3D application. It was created by Sony Computer Entertainment [11] and now it is managed by the Khronos Group [10]. The initial release was in October 2004 and latest release is from August 2008. It was also extended from XML.
Similar to U3D, COLLADA is supposed to enable easy exchange of 3D data among (industrial) applications. Therefore, COLLADA defines an open standard XML schema for exchanging 3D data.
COLLADA is "just" a standardized format of 3D data and not a technology for representing 3D content on the web.

2 3D Web Technologies for The Project 3D Mobile Internet

This part should provide more specific point of view on existing 3D web technologies. All technologies introduced in the previous part are reviewed in the terms of the project's needs and expectations. First, not so convenient technologies are briefly described and for specific reasons rejected. The promising 3D formats for the project are left for the end. For each technology the following properties are described - representation of 3D scene (and its advantages and disadvantages), support of scripting, supported platforms and efficiency of the technology. Other key features may be mentioned.

2.1 U3D and COLLADA

Both are file formats created for easy exchange of 3D data between applications. They are not exactly technologies for rendering 3D scenes on the web. Therefore, neither of them is acceptable for our project.

2.2 VRML

Though standard format and also pioneer between 3D web technologies, VRML has never become as popular among web users as expected. The development was stopped few years ago and therefore, VRML does not keep up with new 3D technologies. The fact, that VRML files are not directly part of the HTML code, also does not make this technology quite appealing for us. Furthermore, the support of media and scripting is also not sufficient for our purposes.
Mainly because of these reasons, the VRML is not convenient for our project.

2.3 O3D

O3D is a quite new technology developed by Google [19].
The content of 3D scene is in a form of an application using JavaScript and it is a direct part of HTML code. The scene could be created either directly or imported. Imported files have to be in COLLADA format. This is quite inconvenient. By creating the 3D scene directly, transform graph and render graph must be created - this is not so convenient either. Furthermore, the scene is not in XML syntax, which is quite intuitive and can be used also by nonprogrammers. It is in a form of an application and even simple static object have to be "programmed" and that is quite

annoying. In other words, the scene representation of this format is not easy to use or understand for "normal" web developers and therefore it will make this format unpopular among them.
Since the whole scene is written as a program in JavaScript, support of scripting is obvious.
O3D is distributed in form of plug-in for web browsers and it is multiplatform.
The software and hardware requirements are quite demanding. Not to mention that there are some issues with certain types of graphics hardware (e.g. Intel GMA 950 or ATI Radeon X1650). Therefore, this format is not convenient for mobile phones.
Another disadvantage is that the 2D and 3D content is not combined and the 3D scene therefore is not really integrated part of the web page.

2.4 WebGL

This technology has the same disadvantage when it comes to the 3D content representation as O3D. The 3D content is also by this format in a form of an application. It is way too complicated. For rendering simple static primitive such as a cube one has to write many lines of code and use various libraries. For web developers without previous knowledge of OpenGL ES 2.0 [14] it is not intuitive and easy to use. The 3D scene is integrated to HTML via Canvas element - which is a part of HTML5. The specification (or better said specification draft) does not mention any direct support for importing 3D models. That would be a big disadvantage of this technology. Also any special support of audio and video files is not mentioned there. In other words, there is no information about multimedia support outside the HTML5 properties. Since it runs in HTML5 element, WebGL has full integration with all Document Object Model (DOM) interfaces.
It is a DOM API and therefore could be used from any DOM-compatible language: JavaScript, Java or Objective C.
WebGL is supposed to be direct part of the most common web browsers - no plug-ins necessary. It is also integrated in WebKit.
WebGL is based on OpenGL ES 2.0 and designed to being very efficient. It should be therefore convenient also for embedded systems (though many users are very skeptic about that).
The big disadvantage of this technology is that is new - there was only one preview release in December 2009. There are many bugs and one cannot tell when stable and really usable release is about to show up. Furthermore, the reaction from users was not as good as expected. The other minus of this technology is the separation of 2D and 3D content. The communication between these two parts is poor in compare with e.g 3DMLW.

2.5 XML3D

The 3D scene is represented by XML syntax and it is direct part of HTML code. The 3D content can be created directly or can imported from some 3D modeling program such as Maya [22] or 3ds Max [23]. The creating of the scene is very intuitive and easy to understand - anyone, who designs web sites should be able to grasp this concept and create at least static 3D scenes. There is support for multimedia and animation.
Since the 3D content is in HTML DOM, the scripting should be possible with JavaScript, Java and also Objective C.
XML3D is right now in form of plug-in for web browsers (for current version of them). In the future however, it is supposed to be direct part of web browsers (similar to WebGL). The agreement was already made with Mozilla [17] and Opera [18] and the discussion with others is in progress. XML3D is also expanding to WebKit [16]. The format is multiplatform and was submitted to W3C [9] for standardization.
One of the goals by designing XML3D was efficiency not only on personal computers but also on embedded systems. Therefore, this technology should be convenient also for mobile phones. However, since there was no release yet, it is hard to tell, if this goal has been achieved or not.
XML3D uses realtime ray tracing for rendering (fully automatic, robust and physically based). However, similar to WebGL, XML3D is a new technology - there has been no release so far. There were promising presentations of this technology with many examples, but how it is going to turn out in practice is really hard to predict now. However, it has predispositions to be more successful and popular among web users than WebGL.

2.6 3DMLW

The 3D content is contained directly in HTML code and is described using XML. The advantages of this approach were already explained by XML3D. The 3D scene could again be created directly or imported. There is support for audio files and pictures, but unfortunately right now there is no support for video files or skeleton animation.
The scripting is of course possible using scripting language Lua.
In the near future, this technology is going to be multiplatform (some test versions are already available). The last stable release however is only for Windows platform. The plug-in is available for Internet Explorer [20], Mozilla Firefox [21] and Opera [18].
The hardware requirements are unfortunately suited rather for personal computers than anything else.
The big advantage of this technology is that it combines 2D and 3D content really well and therefore is really appealing to common web users. Its popularity grows. In the future, after it becomes cross-platform it could become most popular technology for representing 3D content on

the web.

2.7 X3D

X3D is a standard file format. This is a big pro, for this technology. The 3D content is represented directly in HTML using XML syntax. The 3D scene could be either created directly or imported from some 3D modeling program. It is based on well known VRML. There is support for video and audio files, animation, user interaction, navigation, 2D graphics and CAD data.
The manipulation with 3D objects is possible with C, C++ or Java.
X3D is multiplatform and once again it is a standard for representing 3D graphics on the web.
It is also well suited for embedded systems.
Of course, there are also few disadvantages of X3D. Its development lately is rather slow and it does not keep up with some new trends of graphics. The combination of 2D and 3D content is not quite as extensive as it could be - especially in compare with XML3D and 3DMLW. Despite the fact that it is quite old, it is not as spread and popular among users as expected. And since it is really not a new technology, its popularity is probably not about to grow.

3 Conclusion

In the previous two parts some of the 3D web technologies were introduced. In this part, the best technology for the project is discussed and recommended. The technologies are compared according to the way, how they represent the 3D scene, their scripting possibilities, their platform support, their efficiency and their availability.

Three of the 3D web technologies (U3D, COLLADA, and VRML) were already rejected in the previous part and therefore are not mentioned again.

3.1 Representation of 3D Scene

I find the direct XML description of a 3D scene far more convenient as the representation with a program. For the dynamic scenes, one has to use a scripting language anyway, but for simple static scenes it is too complicated to use program representation. Many web developers are not programmers, but they can still create static scenes on their own using simple XML syntax. In other words, every web designer with basic knowledge of HTML and XML can easily put some static 3D content to his web page without the need to program something. This is important, because in that way, technologies based on XML representation have more chance to become really popular and also quite spread among common users. From that point of view, neither O3D nor WebGL is a good choice for representing 3D scenes on the web.

3.2 Scripting Possibilities

Scripting possibilities are more or less the same for all mentioned technologies. A little bit worse is perhaps only 3DMLW which uses Lua scripting language. The best properties in this area has X3D with support of C, C++ and Java.

3.3 Platform Support

Except 3DMLW, all technologies are multiplatform. However, even 3DMLW is about to become cross-platform.
All technologies are also available for the most common web browsers. WebGL is even a direct part of new versions of them (and XML3D should be direct part of them in the near future). For other technologies, plug-ins for web browsers are necessary.

3.4 Efficiency

3DMLW and O3D are currently suited only for PCs. The usability on embedded systems should not be a problem for WebGL and XML3D (at least according to their specifications). X3D is well suited both for PC and embedded systems.

3.5 Availability

X3D is a standard - stable release is available for a long time now. It has all necessary properties. Stable release of O3D is also available and so is 3DMLW stable release (unfortunately this release is not multiplatform). WebGL has been released just recently as a test version, contains many bugs and reaction of users has not been good so far. The stable version and perhaps acceptance among users is expected in the first moths in 2010. However, it all can turn out quite differently. The XML3D technology has the same problem. It is supposed to be released in January or February 2010. If it has all promised features, it could become very popular and could spread really quickly. However, many practical issues can appear after its release.

3.6 Recommendation

After comparing all described technologies, I would recommend X3D for our project. It has all desired properties and it represents standard among 3D web technologies. With time 3DMLW, WebGL or XML3D will be probably more popular and more used as X3D, but right now neither of them has the features necessary for our purposes and is stable enough.
If the stable release of WebGL and XML3D appears soon enough (January or February) and it has all promised properties, I would consider using one of these technologies (and because of scene representation I would prefer XML3D).
Because of the demanding hardware requirements, 3DMLW and O3D are not suited for embedded systems and therefore also for our project.
With time some other technology could me more convenient and X3D could be replaced. In that case, replacement by technologies based on XML would be much easier than by technologies, that represent 3D scene with a program.

References

[1] VRML [http://www.web3d.org/x3d/vrml/]

[2] X3D [http://www.web3d.org/x3d/]

[3] 3DMLW [http://www.3dmlw.com/]

[4] XML3D [http://graphics.cs.uni-sb.de/489/]

[5] WebGL [http://www.khronos.org/webgl/]

[6] O3D [http://code.google.com/intl/cs/apis/o3d/]

[7] U3D [http://www.ecma-international.org/publications/standards/Ecma-363.htm]

[8] COLLADA [http://collada.org/]

[9] W3C [http://www.w3.org/]

[10] Khronos Group [http://www.khronos.org/]

[11] Sony Computer Entertainment [http://www.scee.net/]

[12] 3D Technologies R&D [http://www.3dtech.ee/]

[13] Computer Graphics Group of the Saarland University [http://graphics.cs.uni-sb.de/]

[14] OpenGL ES [http://www.khronos.org/opengles/]

[15] OpenInventor [http://oss.sgi.com/projects/inventor/]

[16] WebKit [http://webkit.org/]

[17] Mozilla [http://www.mozilla.org/]

[18] Opera [http://www.opera.com/]

[19] Google [http://www.google.com/]

[20] Internet Explorer [http://www.microsoft.com/windows/internet-explorer/]

[21] Mozilla Firefox [http://www.mozilla-europe.org/en/firefox/]

[22] Maya [http://www.autodesk.com/maya/]

[23] 3ds Max [http://www.autodesk.com/3dsmax/]

[24] Google SketchUp [http://sketchup.google.com/]

[25] Adobe Flash [http://www.adobe.com/products/flash/]

[26] OpenGL [http://www.opengl.org/]

[27] DirectX [http://www.microsoft.com/games/en-US/aboutGFW/pages/directx.aspx]

[28] Apple [http://www.apple.com/]

[29] Intel [http://www.intel.com/]

[30] Right Hemisphere [http://www.righthemisphere.com/]

[31] The Boeing Company [http://www.boeing.com/]

[32] Adobe Systems Inc. [http://www.adobe.com/]