Comparative Study of Web3D Standard Format to Determine the Base Format for A Web3D Framework

Mursid W. Hananto
Department of Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
mursid.wahyu.h@mail.ugm.ac.id

Ahmad Ashari
Department of Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
ashari@ugm.ac.id

Khabib Mustofa
Department of Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
khabib@ugm.ac.id

***Abstract*— With the current Web3D document format, users are forced to choose certain document formats to use, either during development with a particular tool or when it will be displayed in a browser. Only one format that can be processed by any browser at one given time. This raises the main problem of not allowing users to display a variety of objects with different formats in their browser. For this problem, a Web3D framework can be the solution, as it will provide format conversion for the browser. The conversion itself requires an appropriate base format as the conversion goal. Since there are many formats that have been implemented by users, a comparison has to be done for the purpose of choosing the suitable format.**

In this study, comparisons have been made to obtain some information. The information required is the complexity of each document in describing a 3D object in the browser, as well as the performance of the particular format. Web3D formats compared in this research are the standard ones: VRML and X3D. Various specific description of object formation have also been selected as sample representation for each format.

Based on comparisons in the representation information of each standard format, X3D is the more suitable format for this need. As a standard format representation, the results obtained can be used for further comparisons with non-standard or proprietary formats. This information is needed to determine the final base format for the framework to be developed in subsequent research.

Keywords*— *Web3D, VRML, X3D, comparison, format, standard

I. Introduction

The need to display more than one format at a time is based on the desire to shorten the development time of the Web3D site, especially those with a world that has high object complexity attributes with large quantity of constituent elements and large viewing area. Creation of such Web3D world is very difficult and time consuming. The choice of solution is to utilize the principle of reusability in the form of the use of existing components to provide convenience to its users [1]. The component used can be either an individual object or a world, which is in a separate place. The world from various sources is then called by the main world that unites it so that it appears together all at once in a browser. In Web3D, this approach is referred to as a distributed world [2].

In a conventional web as per W3C standards, a page view in a browser can be built from an HTML document. When a user want to display the contents of another HTML document, the document should be displayed in a different window. Sometimes a developer chooses a solution of using frames that have been included as HTML component since version 4.0 [3]. Distributed world in Web3D makes it possible to bring up the contents of more than one source document in a browser window without using frames. It's just that distributed world only applies if all documents of the world have the same format. Thus, a user can simply create a major world, then invoke various other worlds that already exist and build by different creators, but all the worlds called should have the same format.

References to the world or scenery as well as external individual 3D objects can not be made to documents with a different format than the main document format. It is also presented in [4] which made research to display 3D graphics in various formats by converting them into a suitable format. In Web3D, users are forced to retrieve documents manually from the site containing the required objects. After that it's converted to a format that matches the document format of other objects that users want to display together in the browser, although there are potential problems in converting documents from one format to another [5].

The concept of the solution being researched is a framework that has the ability to simultaneously display objects according to descriptions on documents from multiple sources in different formats. The research in [6] also have a similar purpose, but the merged results are displayed in a certain application as the final viewer, and this application takes input of 3D objects individually. Proposed framework in [7] requires the use of a standard browser as its main viewer component. Therefore, a special browser is not built for this purpose so that users can directly use their own browser as they wish and have been familiar with it. For compatibility reasons, the browser must comply with W3C standards [8]. Because it uses a widely available browser, this concept requires the existence of a base format that serves as a basis for uniform formats that are used as inputs to be displayed as an integrated

Mursid W. Hananto is also with Universitas Ahmad Dahlan, Yogyakarta, Indonesia (e-mail:mursid@is.uad.ac.id).

978-1-5386-0549-3/17/$31.00 ©2017 IEEE

view in users' browser. In addition, the base format must be a format that has been widely accepted and used in various Web3D implementations.

Of the many formats, there are two Web3D document formats that have been officially defined by the Web3D consortium. The formats are VRML [9] and X3D [10]. Both are the most widely used 3D formats [11] and have similar main characteristics that require plug-ins, and save the object description document in the form of a text file. Since the base format is only one, then one of the two standard formats considered more efficient is selected. The efficiency of the base format can be seen from two aspects, namely performance when displaying a Web3D world, as well as from the complexity of documents that are formed to display 3D objects in the browser. Because the desire to add realism and design in 3D graphic display is increasing [12], the format must be able to anticipate the effect on display rendering performance. This includes when using certain techniques to improve display performance when the object described is very complex, for example by using LOD or Level of Detail [13].

VRML has been widely adapted for graphical representation of 3D objects over the web [14]. Earlier implementation VRML was difficult since there were no support from commercial 3D object-making software, but now there are several tools that can be used to help create 3D models in this format [15]. At first, VRML was about to be selected as the base format for the framework to be studied due to the popularity and number of users of this format. Since X3D is the newer standard format compared to VRML, it is necessary to perform a comparison in order to select the more suitable base format. Comparisons were performed on the complexity of the formation of objects in the document as well as the results. The results were compared on various browsers with a combination of various plug-ins is done.

II. WEB3D

Web3D is a term that describes a programming or descriptive language that can be used to provide interactive objects and 3D worlds over the Internet. Web3D includes open language such as VRML, Java 3D and X3D - as well as any proprietary language that has been developed for the same purpose under the umbrella of the Web3D consortium. The standards set by the Web3D consortium are open. Web3D open standards have also strong relationships with other standards for multimedia [16].

Virtual Reality Modeling Language (VRML) is a file format for describing interactive objects and 3D worlds. VRML is designed for use on Internet, Intranet, and local client systems. VRML is also intended to be a universal exchange format for integrated multimedia and 3D graphics. VRML can be used in a variety of applications such as engineering and scientific visualizations, multimedia presentations, entertainment and education, web pages, and shared virtual worlds [17].

VRML is an international standard file format based on ISO/IEC 14722 [9], to describe interactive 3D multimedia on the Internet. The VRML 1.0 specification is issued by Silicon Graphics Inc. company, and is based on the Open Inventor file format. The second release of VRML has gained a huge addition in terms of interactivity capabilities. This second generation was designed by the VRML team of Silicon Graphics Inc. with contributions from Sony Research, Mitra, and more. VRML 2.0 has been observed by the VRML discussion group through moderated e-mail (www-vrml@vrml.org) and then adopted by many companies and individuals. In December 1997, VRML97 replaced VRML 2.0 and was officially defined as an international standard ISO/IEC 14772 [18]. Fig. 1 illustrates a conceptual model of a VRML browser.

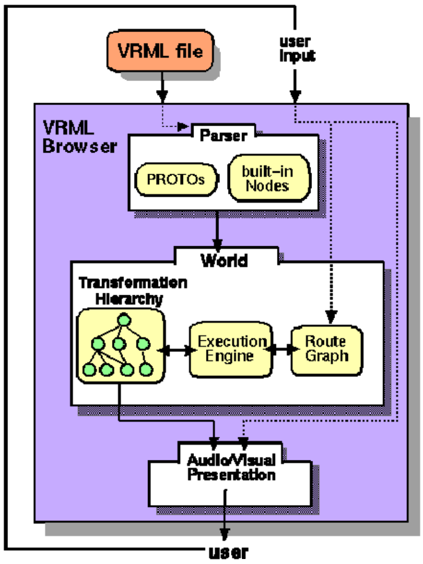

Fig. 1. Conceptual model of a VRML browser [19].

X3D is a royalty-free open standard file format and run-time architecture to represent and communicate with 3D and objects using XML [20]. This format is a ratified ISO standard and provides a system for storing, retrieving and displaying real-time graphic content that has been prepared in the application, all in an open architecture to support multiple domains and user scenarios [21]. Fig. 2 illustrates the X3D system architecture.

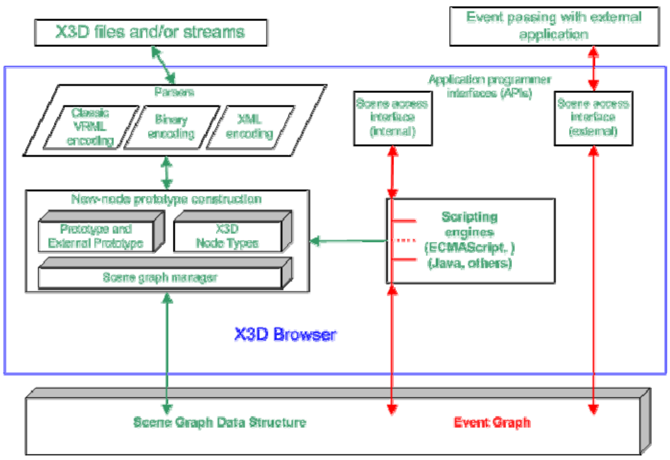

Fig. 2. X3D architecture [22].

X3D has a set of customizable features that can be used as components for use in scientific engineering and visualization, CAD and architecture, medical visualization, training and simulation, multimedia, entertainment, education and more. The development of real-time 3D data communications across all forms of application has evolved from the beginning as Virtual Reality Modeling Language (VRML) with the more mature and refined X3D standards [21].

III. COMPARISON

A. *Provision of Document Writing*

Both VRML and X3D formats should describe headers and content in their documents. In VRML, the header row starts with writing the #VRML V2.0 utf8 text, whereas in X3D the header row starts with the text <?xml version="1.0" encoding="UTF-8"?>. Because X3D uses XML description language [23], it is possible in X3D to describe a reference to which specification and which schema the document refered to. This is not implemented in VRML, where after starting with the document identifier line, the next section directly describes the contents of the document.

Both VRML and X3D should also specify each attribute for each node to be portrayed in the browser. VRML is made up of nodes, which contain mathematical descriptions of 3D points, lines, surfaces, text, strings, and solids [24]. In general, the node can be a simple representation of a particular shape, as well as a description of complex objects. Complex objects can be formed from a variety of simple objects in various ways of formation, as well as describe them by coordinates.

When using a combination of objects, it will get some description of a child object that is covered by the name of a larger object. But apparently based on the initial experiment, this method did not has to be done, because it could be a complex object constructed by a collection of objects, but the object does not has to be the child of the larger description. This is because a node or object has 3 types of definitions of transformation i.e. position, direction, and scale. When the object definition causes the object to attach itself to another object even though it is not a child node of the larger node or object description, this node will give the scene graph a new form on the object it is attached to.

Because the two formats are based on the same basic needs with the difference in the further development, then for the selection of the object formation pattern is based on the description of the same node and contained in each format. Each format has a basic node description for constructing simple objects but it is possible to get changes to the property and its value. The base node is also called primitive. As both standard formats are defined by the Web3D consortium, the basic nodes or fundamental objects of the two formats are essentially the same. Therefore, the same basic nodes of the two formats, namely Box, Cone, Cylinder, and Sphere have been selected for the first phase of comparison.

Each primitive or base node is created based on its default default value. In addition to not changing default values, additional properties are not granted, and Material nodes are not altered so only use default values. All other descriptions either in the entire scene graph or non-default world are not added, so objects rendered in the browser are really just default objects without any manipulation. All browsers on the computer for testing are in default. Finally, no additional code is inserted in any object description document. This is done so that comparisons only look at how each encoding is done for the primitives.

Nodes with intermediate complexity are created based on a combination of nodes in a special node, the Group node. For experimental purposes, 2 Group nodes are built using some basic nodes to form new nodes. High-complex nodes are based on coordinate descriptions. Objects with descriptions and same scene graph scenes are created in both formats to allow for comparison.

B. *Sample Comparison*

The VRML format uses a way of description of objects that are only used in this standard. The encoding form is specific and does not allow the use of other descriptions that do not conform to predefined standards, applicable from VRML 1.0, VRML 2.0, or VRML97. On the other hand, X3D uses XML encoding. XML is the standard that Web consortium has set as a way of describing information structured on the internet. XML has been widely used for various purposes especially for data transfer. Due to the widespread use of XML, its use in Web3D format will facilitate its reception so it can quickly be implemented into code to describe scene graphs.

For comparison to show the complexity value of each format, the X3D format document does not use the classic VRML encoding. Thus the X3D node for experiment in comparison is only made in XML encoding. Each object description document is created with a simple text editor using ANSI text encoding, the goal is to minimize the contents of the document so that comparisons can actually be performed only on the node description only. Compression of documents in both formats is also not executed.

Generally based on the encoding of each format for all primitive base objects with default values, the only significant difference is only visible in the header. The header is an important part because it is used to notify the browser of the type of document that the browser will process before it is displayed. The header for the VRML document is very short, while the header for the X3D document is longer because it must define references and profiles for validating the way the document is written. The X3D format also requires defining the Scheme before defining the object and its attributes to be described. For simple objects or nodes, the VRML file size can be smaller when compared to files containing X3D documents.

Another thing to be observed is the use of a description style object that is similar to the C language in the VRML document format. This writing style potentially generates more typographical errors when editing descriptions, especially since the writing of descriptions in the document is case-sensitive. On the other side, the style of writing markup language for web documents are used in X3D format. Web users who are familiar with HTML especially version 4.0 and later will be greatly helped by how X3D documents are written.

The next thing to look at is the complexity of documents produced by both VRML and X3D formats. Calculation of document complexity can not be done by calculating the complexity of lines of code as in [25] and [26]. This is because all object descriptions of both VRML and X3D formats are linear, so the contents of the document lines of each format do not have branching or selection of conditions as in the line of program code. X3D uses a similar XML writing pattern and is developed from VRML. It is therefore chosen how to calculate complexity through elements and attributes definitions/declarations, elements and attributes group definitions/declarations, and other definitions including user defined and built-in simple type and complex type definitions [27].

Prior to the calculation of complexity, careful scrutiny is done on the code line of each document from each format that gives the same display results in the browser. Care is done on how many command lines should be assigned to each format in order to produce a view.

The comparison of the number of commands used to describe the node/object in a scene graph by counting the number of lines of code for each document can be summarized in Table I. The line counts of both formats are listed in the table to show the results on the basis of each comparable document. Table II shows the results of complexity of document samples from both formats.

TABLE I. LINE COMPARISON

Objects	VRML	X3D
Sample node: Box	5	14
Sample node: Cone	5	14
Sample node: Cylinder	5	14
Sample node: Sphere	5	14
Sample node: Group 1	16	28
Sample node: Group 2	32	46
Sample node: Complex 1	23	18
Sample node: Complex 2	19	17

TABLE II. OBJECT DESCRIPTION COMPARISON

Objects	VRML	X3D
Sample node: Box	4	5
Sample node: Cone	4	5
Sample node: Cylinder	4	5
Sample node: Sphere	4	5
Sample node: Group 1	14	19
Sample node: Group 2	42	41
Sample node: Complex 1	42	32
Sample node: Complex 2	37	31

As seen on Table I, the amount of lines of codes in X3D documents used by the samples are higher, but as the sample nodes are getting more complex, the differences of recorded values in line amount are getting smaller. When it comes to complexity values, X3D sample documents used in the tests provided smaller numbers when they were used to display the more complex object, except for only 1 node which is a group node. This group node is actually a complex model constructed from many simple objects. On the contrary, as the constructing single objects became more complex in the other group node, the resulting complexity amount is smaller.

C. Sample Test

After comparing the forming lines of documents in both VRML and X3D formats, it is necessary to know the result when documents from both formats are displayed in the browser. For that purpose it is necessary to test the sample documents that have been compared in the previous section. The test was done using 4 browsers and 2 plug-ins/viewers.

Samples were taken from documents that use Group nodes and single complex nodes. No samples from primitive types because they are too simple and concise, and thus less likely to produce significant differences. Both document types of each formats are used as a short test material to find out how each document will perform when displayed in the browser. The following are test results on all documents. The test is performed by taking the average value of each selected navigation mode, and the recorded value is measured in fps (frame per seconds). Each test was done 20 times, and each single test took 31 seconds (the first second were not counted), the value recorded were from every 3 seconds. The results from walk mode, pan mode, and observe mode tests are shown in Table III, Table IV, and Table V respectively.

TABLE III. RESULT FROM THE TEST, WALK MODE

Objects	VRML	X3D
Sample node: Group 1	42,2 fps	42,3 fps
Sample node: Group 2	36,7 fps	35,1 fps
Sample node: Complex 1	14,1 fps	16,6 fps
Sample node: Complex 2	11,8 fps	15,7 fps

TABLE IV. RESULT FROM THE TEST, PAN MODE

Objects	VRML	X3D
Sample node: Group 1	41,6 fps	42,9 fps
Sample node: Group 2	33,2 fps	32,4 fps
Sample node: Complex 1	13,5 fps	13,1 fps
Sample node: Complex 2	11,1 fps	12,9 fps

TABLE V. RESULT FROM THE TEST, OBSERVE MODE

Objects	VRML	X3D
Sample node: Group 1	43,5 fps	43,8 fps
Sample node: Group 2	36,4 fps	39,2 fps
Sample node: Complex 1	16,5 fps	16,3 fps
Sample node: Complex 2	15,3 fps	15,9 fps

In walk mode test, the results are generally not much different, except for the Complex2 node. As seen on Fig. 3, this node is basically a highly detailed 3D model ornamental iron railings which has complex basic shapes mimicking the real objects in the real world. Scale and size of this model are also taken from the original object. It also contains much more complex vertex coordinates compared to other samples nodes. In pan mode and observe mode tests, the resulting values are also not much different, but further scrutiny from overall test results shows that the X3D provides higher average results more often than VRML.

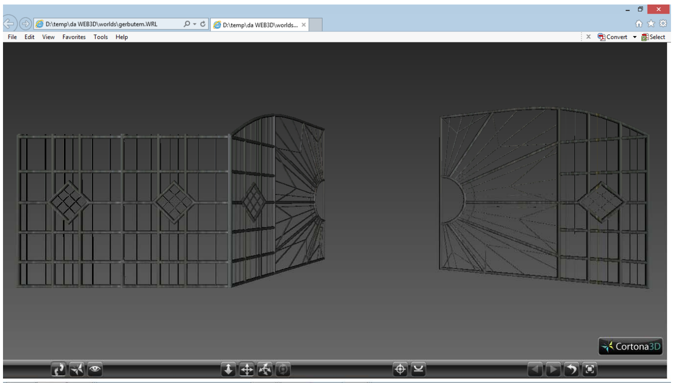

Fig. 3. Sample Object for Complex2 node.

IV. DISCUSSIONS

From the comparison of the encoding method, according to Table 1, it can be concisely to know the comparison between VRML and X3D using some sample documents that contain both simple and complex nodes. Based from the previous tests, X3D requires more lines of code to build simple or primitive objects. This is due to the addition of several lines to comply with XML encoding which is the basis of how to write code in X3D. In addition, the way X3D writes code that adapts the standard tag writing style to web documents requires clarity of the opening and closing parts of the document. It also provides additional lines of code into X3D documents even though basically the existing line of elements can be aligned, with the impact that developers will have little difficulty following the flow of the document. Beyond these additions, the actual number of lines of code from X3D is not much different from VRML.

In the comparison of Group node code line section, it appears that the number of rows in X3D documents have more lines than VRML documents. But unlike the primitive part, the number is no longer up to more than two times the lines of VRML code. The Group node samples in comparison are basically a combination of some primitives which then added with additional properties along with attribute values to modify the appearance of the object, either individually or in whole. The primitive is then positioned and aligned so that it entirely forms a single object. Site visitors will see all these objects in their browsers as visible and explorable objects, and can interact further when enclosing the sensor code to detect the user's response.

In a single complex node section, there is a tendency of X3D that is able to shorten the code so that the overall number of lines of code in the document becomes less than VRML. From this it can be said that based on the comparison of the samples made for this study, if the code used to describe the object is increasingly complex then the difference between the two document formats is greater.

In all forms of documents that describe the data, the number of lines of existing code can affect several things, including the loading time to memory, the amount of memory needed, and how long the process required to do the translation to be understood by the machine. Additionally, the line of code also affects the file size which determines the transfer time from the server computer to the user's computer as it passes through the computer network. In order to find the base format for the framework, the shorter code line makes it easier to build new documents to display conversion results than other formats. In addition, when viewed manually, the compact code allows developers to build 3D objects faster, and when the process is automated the system can generate code in shorter time.

Based on comparison results, VRML tend to provide more complex code and complexity in increasingly complex documents while X3D raises better complexity calculation results. In reality, recent scene graphs have become increasingly complex and rarely contain primitive or primitive assemblies, therefore the more compact code and lower complexity counts for complex objects in X3D can be an important point. X3D format is a not a process description so it is natural that its content can not be directly understood by users in a single reading. However, such a rigid XML-style structure will make it easy to recognize the parts of the document that are needed, especially when changing the shape of the data in the conversion process.

To obtain informations on the performance of documents when displayed, comparisons are made when the code is realized to be an object in the browser. Comparison is done by performing 3 common navigation modes performed in Web3D, i.e. walk, fly, and observe. Each navigation mode can have various motion combinations, but the one used in comparison is the default mode pair each provided by the Web3D object viewer in the browser. Based on the experimental results for comparison, the responses to interactivity were not different in overall objects in both VRML and X3D versions, and the values obtained in the test to obtain frames per second were quite variable. The average result of 4 browsers combined with 2 different plug-ins does not show a particular pattern, which indicates that based on the samples used, both have similar performance although the X3D tend to provide a slightly higher numbers on most of the results.

V. CONCLUSION

The comparison results have shown some annotations between the two formats compared in the study, so they can be used to indicate the choice of standard formats to be used as the base format in subsequent research of Web3D frameworks with the main capability of viewing multiple formats from different sources. Based on the comparative experiments conducted, it can be concluded that the X3D format is more suitable to be used as the base format compared with VRML. In addition, its proximity to XML will make it easier to build the framework which based on this research.

ACKNOWLEDGMENT

The authors would like to thank to the Head of Multimedia laboratory, Department of Information Systems at Universitas Ahmad Dahlan (UAD) for giving the permission to carry out the experimental research. The authors would also thank to the Head of SKJ Computer Laboratory, Department of Computer Science and Electronics at Universitas Gadjah Mada (UGM) for giving the permission to use the facility to complete the final report.

REFERENCES

[1] R. Anguswamy and B. Frakes, “Study of factors affecting the design and use of reusable components”, Proceedings of the 7th International Doctoral Symposium on Empirical Software Engineering (IDoESE), Lund University, Sweden, September 2012.

[2] M. Pesce, VRML: Browsing & Building Cyberspace. Indianapolis: New Riders Publishing, 1995.

[3] D. Stotts, and J. Navon, “Model checking CobWeb protocols for verification of HTML frames behavior”, Proceedings of the 11th International Conference on World Wide Web, Hawaii, USA, May 2002, pp. 182-190.

[4] R. B. Berthelot, J. Royan, T. Duval, B. Arnaldi, “Scene graph adapter: an efficient architecture to improve interoperability between 3D formats and 3D applications engines”, Proceedings of 16th International Conference on 3D Web Technology, Paris, France, June 2011, pp. 21-29.

[5] The National Archives, “File Format Conversion - Version:1.2”, Kew, Richmond, Surrey: The National Archives, 2011, http://www.nationalarchives.gov.uk/documents/information-management/format-conversion.pdf.

[6] R. B. Berthelot, J. Royan, T. Duval, B. Arnaldi, “3DFC: a new container model for 3D File formats compositing”, Proceedings of the 17th International Conference on 3D Web Technology, Los Angeles, California, August 2012, pp. 27-35.

[7] M. W. Hananto, “Universal framework for displaying multi-format Web3D worlds”, Proceedings of International Conference on Culture, Communication, and Multimedia Technology (ICON C-COMET 2014), Universiti Utara Malaysia, Sintok, Malaysia, April 2014, pp. 163-172.

[8] W. Seltzer, “World Wide Web Consortium (W3C) standards for the open web platform”, in Open Source, Open Standards, Open Minds Conference Proceedings, Washington DC, April 2016.

[9] Web3D Consortium, “The Virtual Reality Modeling Language”, Web3D Consortium, 2006, http://www.web3d.org/documents/specifications/14772/V2.0/index.html.

[10] Web3D Consortium, “Extensible 3D (X3D) Part 1: Architecture and base components – Introduction”, Web3D Consortium, 2013, http://www.web3d.org/documents/specifications/19775-1/V3.3/index.html.

[11] Web3D Consortium, “X3D & VRML, The Most Widely Used 3D Formats”, Web3D Consortium, 2017, http://www.web3d.org/x3d-vrml-most-widely-used-3d-formats.

[12] S. Jeschke, M. Wimmer, W. Purgathofer, “Image-based Representations for Accelerated Rendering of Complex Scenes”, STAR (State of the Art Report) in the 26th annual conference of the European Association for Computer Graphics (EUROGRAPHICS 2005), Trinity College Dublin, Ireland, September 2005.

[13] W. Pasman and F. W. Jansen, “Scheduling level of detail with guaranteed quality and cost”, Proceedings of the 7th International Conference on 3D Web Technology, Tempe, Arizona, USA, February 2002, pp. 43-51.

[14] L. W. Ming and I. Gibson, “Specification of VRML in Color Rapid Prototyping”, International Journal of CAD/CAM, Vol. 1, No. 1, pp. 1-9, 2002.

[15] G. Pomaska, “Implementation of Web 3d Tools for Creating Interactive Walkthrough Environments from Building Documentations”, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS), Vol. XXXIV, Part 5/W12, (ISPRS WG V/4 and IC WG III International Workshop on Vision Techniques for Digital Architectural and Archaeological Archives), Portonovo-Acona, Italy, July 2003, pp. 268-272.

[16] L. Chittaro and R. Ranon, “Web3D technologies in learning, education and training: Motivations, issues, opportunities”, Computers & Education, Volume 49 Issue 1, pp. 3-18, August 2007.

[17] R. Carey, G. Bell, C. Marrin, “ISO/IEC 14772-1:1997 Virtual Reality Modeling Language (VRML97)”, The VRML Consortium Incorporated, 1997, http://www.vrml.org/Specifications/VRML97.

[18] VRML Consortium, “The Virtual Reality Modeling Language Specification Version 2.0”, VRML Consortium, 1996, http://vrml.sgi.com/moving-worlds/index.html.

[19] VRML Consortium, “The Virtual Reality Modeling Language - Clause 4: Concepts”, VRML Consortium, 1997, http://www.vrml.org/Specifications/VRML97/part1/concepts.html.

[20] D. Brutzman and L. Daly, X3D: Extensible 3D Graphics for Web Authors. San Fransisco, CA: Morgan Kaufmann Publishers, 2007.

[21] Web3D Consortium, “Getting Started with X3D”, Web3D Consortium, 2017, http://www.web3d.org/getting-started-x3d.

[22] Web3D Consortium, “Extensible 3D (X3D) Part 1: Architecture and base components - Clause 4: Concepts”, Web3D Consortium, 2013, http://www.web3d.org/documents/specifications/19775-1/V3.3/Part01/concepts.html.

[23] M. Gelautz, M. Brandejski, F. Kilzer, F. Amelung, “Web-based visualization and animation of geospatial data using X3D”, Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS '04) Volume 7, Anchorage Alaska, September 2004, pp. 4773-4775.

[24] S. Yu, “Slicing Algorithm of VRML Model in Rapid Prototyping”, Journal of Modeling and Optimization, Volume 7, Number 1, pp. 45-49, 2015.

[25] T. J. McCabe, “A Complexity Measure”, IEEE Transactions on Software Engineering, Vol. SE-2, No. 4, December 1976.

[26] R.S. Pressman, Software Engineering: A Practitioner’s Approach 5/e. New York: McGraw-Hill, 2000.

[27] D. Basci, S. Misra, “Complexity Metric for XML Schema Documents, Proceedings of the 5th International Workshop on SOA and Web Practices, 2007, pp. 1-14