Journal of Theoretical and Applied Information Technology
28th February 2019. Vol.97. No 4
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RESOURCE MANAGEMENT AND CONVERSION PROCESS IN MULTI-FORMAT DISTRIBUTED WORLD WEB3D FRAMEWORK

[1]MURSID WAHYU HANANTO, [2]AHMAD ASHARI, [3]KHABIB MUSTOFA

[1,2,3]Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia

[1]Department of Information Systems, Universitas Ahmad Dahlan, Indonesia

E-mail: [1]mursid.wahyu.h@mail.ugm.ac.id, [1]mursid@is.uad.ac.id, [2]ashari@ugm.ac.id, [3]khabib@ugm.ac.id

ABSTRACT

Current Web3D no longer only contains simple objects or sceneries with low complexity. Even without the addition of certain application features, the content of a Web3D site can be so complex that its development can take a long time. Development time can be reduced by using contents or complete world from other sites, so users can directly display views that integrate various content according to the distributed world principle in Web3D. However, this principle cannot be applied to contents with different document formats, as general browsers can only display one format at a time. The proposed solution is a framework which contribute to the distributed world principle by adding the ability to combine two or more format in a single world. This Framework allows users to combine contents from multiple source sites and formats at once. To overcome the major weakness from the browsers, all document format originating from the source worlds need to be uniform. A series of resource preparation followed by conversion process have been developed so that all content has the same format. Prior to conversion, resources from source worlds were managed locally. The process model of the framework has been implemented and able to demonstrate the capabilities needed to convert two or more worlds with different formats to be displayed in one browser window, thus users can interact with the combined worlds as if they were navigating in a single large world.

Keywords: *Multi-format, Management, Conversion, Framework, Web3D*

1. INTRODUCTION

Creating a 3D model takes a long time and require high costs [1], especially if the 3D model has high detail [2] to be used to build a large-scale world and have a complex appearance. Although the development can be accelerated with tools such as using 3D modeling software [3], it will still take a long time because of the high level of difficulties in creating complex 3D models and their properties for a world [4]. There are several specific authoring tools for Web3D, such as tools for VRML format [5] or for X3D format [6], Web3D application as a 3D authoring tool for mobile systems [7]. Vendors also provide commercial authoring tools, including RapidAuthor [8] and BS SDK [9]. However, special authoring tools does not make the job easier, because mastering these tools also requires lots of time and efforts. Following the creation of 3D models, more time is needed to add specific features into the models if the needed product were not just a 3D graphical display and its interactions, e.g. using animation on objects rendered on the client side [10] all the way to a combination of features that make the world a complex web-based interactive application, such as applications for training of automatic weather stations [11].

To shorten development time, basically web developers can adapt the concept of modularity by utilizing components that have previously been built [12]. However, in Web3D this is done by using world components in the form of 3D models refered directly from a different source on the internet, or even using the complete world from one or more other Web3D sites to be displayed together like a single world. Thus, users can form a world that literally consists of several worlds and displayed together in one browser view, utilizing the distributed world principle [13]. With this way, users do not have to build their own complete world as a whole, but they can use multiple worlds as well as existing ones to compose a larger and more complex world, as illustrated in Figure 1. This concept actually allows users to seamlessly explore a single, very large world which is a combination of multiple separate worlds from different sources. If users can navigate between world uninterrupted, then they do not need to move from a site to other sites when they want to explore multiple worlds in their browser.

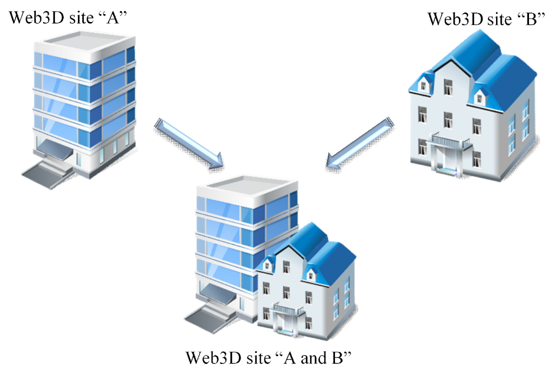


Figure 1: Combining Worlds for A Larger Single World

This is not a problem when all the world displays its objects with the same document format. In reality, various developers build their respective worlds using different document formats. There are various Web3D document formats that continue to grow due to the increasing support of computer hardwares and browsers which made 3D graphics have an increasing role in the web environment [14]. Application of various document formats for similar purposes can be seen in several applications. For example, virtual campus can be built in VRML format [15], or in X3D format [16], or in WebGL format [17]. But none of [15], [16], or [17] was discussing about the probability of integrating the similar worlds in the browser.

The Web3D Consortium has set the standard document formats namely VRML and X3D [18], and each format has obtained a standard number namely ISO/IEC 14772-2: 2004 for VRML [19] and ISO/IEC 19775-1: 2013 for X3D [20]. However, not all developers have the willingness to use them for various reasons. This raises difficulties for users who want to get a visual display which is a combination of several resources at once to be displayed into a world. They are forced to build their own objects for the needed world, or if the objects already exist then they will take them from various sources and converts the components needed or even all objects in the world to the required format. After doing the format conversion, the components then manually combined one by one with other components in a long process.

A mechanism is needed to manage content obtained from one or more source sites, convert the obtained objects, make modifications as needed, make adjustments between content, then compile it into a complete final world before it is displayed in the browser. This final world will only have a single format so that it can be displayed in any browser as long as the browser has support for the particular format.

This paper presents the first two elements of a framework that can merge two or more worlds from different source sites and in different formats according to those supported by the system, then the results are displayed in the user's browser so that users can explore the combined worlds as if they explore a single world. The implementation prototype shows that the framework has succeeded in achieving its objectives so that a combination of certain worlds in the X3D, VRML, and X3DOM formats can be displayed simultaneously in a compatible browser.

2. RELATED WORKS

A complex and large world has many constituent elements, formed in a hierarchical structure [21]. In addition to 3D objects, a world can contain images or graphics, as well as other elements including multimedia. In a world, these elements are made in the form of files that make it easy to create, use, or modify them individually. In accordance with the format used, the file is a document with content that has been formed according to the format. This file is then used by the main world to build the entire world that will be shown in the browser. The browser reads according to the format used and translates into 3D objects which are then rendered on the screen.

When the browser reads a world component files, the browser only reads according to the format that is currently active. If a file with a format that does not match the active format were found, the file with different formats will cause errors when processed. Thus, the world can only be composed of one document format, so it is not possible to display more than one document format to describe 3D objects in one browser view without a certain mechanism. Mixing a different format into a particular format file is also not possible, because it will cause errors with the result that the world will not be able to be displayed at all.

Several studies to manipulate contents in the world have been carried out, including creating JavaScript-based libraries in HTML5 to progressively renders 3D objects in browsers [22], progressive rendering using WebGL [23], and research to make a combination of 2D and 3D contents to be displayed in 2D or 3D display mode [24]. Other than that, there were also studies that aim to facilitate users with any device to view 3D content via the internet, since the rendering loads were placed on the server and the display were sent as image streams via a Java-based client [25]. However, combining more than one world from different sources and formats has never been done before since each format has its own scenery arrangements to create the world, and they are not compatible to each other. Incompatibilities can be seen even on those which originated from the same source, as seen on VRML and X3D which were created by Web3D Consortium. Although X3D documents can be written in classic VRML styles [26], these documents are still X3D documents that can not be displayed in VRML browsers.

There were studies to show that more than one 3D graphic format can be displayed in an application specifically made for that purpose [27]. The solution given in the said study was to wrap different 3D document formats into one known format. In subsequent studies, different formats were then packed into a data container called 3DFC before being sent to the rendering section [28]. The solution built was not specifically for Web3D documents even though it can accept X3D files, and the results were not intended to be directly used in web browsers that were currently widely used. The system processes individual objects, so the solution was not intended for a scenery that includes a collection of objects that form a world. The overall application also requires the availability of special libraries that were in accordance with the document format so that it can work to wrap and adapt a particular document format.

Web users generally use browsers to surf the web, so the solution resulting from a combination of documents will be better if it can be displayed in a browser. However, creating a new browser that has the ability to read various types of formats can not be done in a relatively short time. When using the current browser, only one Web3D format that can be displayed at a time. This applies to specific Web3D browsers from i3D [29] to BS Contact [30] which is one of the current Web3D browsers. The same thing applies to conventional browsers which is actually not intended to display 3D graphic contents for the web.

Based on the limited ability of browsers which can only display one format at a time, the basic idea for displaying more than one world that has two or more format originated from several source into a browser was to make all documents have the same format. This uniformity can be done by converting all other format to the base format. Base format is the one intended as destination format for all document. In terms of conversion, several studies have been conducted including universal converters for 3D graphics [31], management in supported formats [32], and interfaces for displaying converted documents from various different original formats [33]. Besides not being devoted to Web3D documents, the resulting service were used to convert documents individually, not to convert a scenery consisting of many 3D objects with their respective properties and combine them with other sceneries.

3. PROPOSED FRAMEWORK

The proposed mechanism was arranged into a framework to facilitate its implementation. This framework has been proposed in [34], however, there was a problem pertaining the difficulty of implementing the necessary library in the format conversion section in the conceptual framework being researched. Proposed on-the-fly conversion was very difficult to be implemented, therefore a different approach has to be used in the form of localizing resources to the user's computer. The framework in the proposal was modified into four main parts, namely resource management; format conversion; attribute adjustment; and display preparation, as shown in Figure 2.

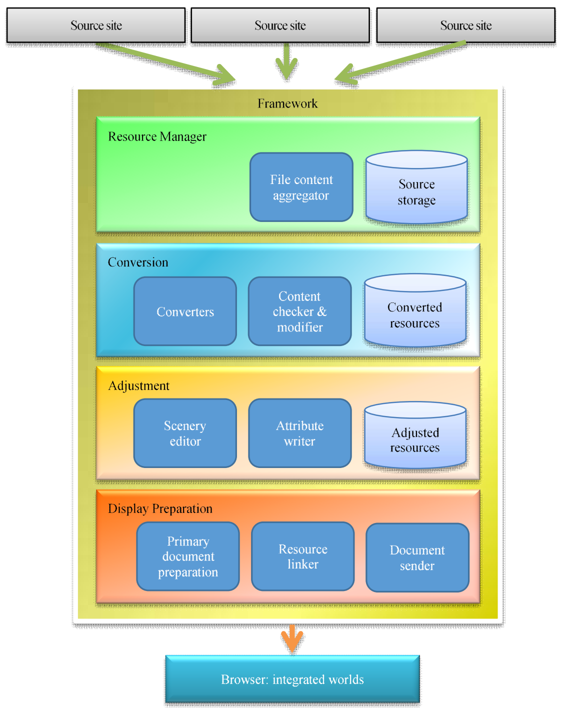


Figure 2: Proposed Framework

This paper discusses the first two elements of the framework designed to prepare the source worlds to be integrated. They were crucial parts since in these series of processes the uniformed files that needed to be integrated will be obtained. Thus, the first two elements of the framework can actually produce component files which can be put together so that the preliminary results can be shown. However, this condition only applies to content from source worlds whose contents were already adapted. If there was any source world which have a display pattern that was different from the aspect of location, direction, and size, then it requires certain adjustment and preparation which were not yet discussed in this paper.

4. BASE FORMAT SELECTION

According to the proposed framework, a final format is required to be displayed by the browser. We call this format as the base format, because all other formats will be converted to this format. This format should be a format that is accepted wherever possible by as many web browsers that are currently available, as the world combination generated by the framework is intended to be displayed in a web browser that has been installed on the user's computer without any changes.

In [35], a comparison was made to determine the base format based on the standard X3D format currently available. However, using X3D will potentially create new problems, especially when the implementation of the framework will be used directly by users. The use of the X3D format requires the installation of a plug-in [36] to the user's computer system so that documents with X3D format can be displayed in the browser. Due to the need for plug-ins, security and/or compatibility issues with browsers might emerge [37]. In addition, the use of plug-in limits the range of users because X3D plug-in are not always available for all platforms or browsers, even though X3D has high performance potential since plug-ins can utilize hardware and graphic drivers [38].

Several other studies related to the selection of Web3D document formats give rise to an alternative to X3D. In [39] and [40], data was taken in CityGML document format to be used as data for 3D models that use X3D format, then sent to WebGL compatible browsers to display the final results using the X3DOM format. In this study, X3DOM was chosen mainly because it allows the result to be displayed without plugins. X3DOM itself is a framework as in [41] and [42], which allows users to display the world in a compatible browser without having to install a plug-in. X3DOM uses DOM in HTML5 and utilizes WebGL through JavaScript library [43]. Elimination of the need to use plug-ins gives rise to the ease of use factor. It also widen the range of usage of framework implementation to integrate different world formats. The absence of plug-ins also minimizes compatibility issues with browsers. Based on the survey, many users want to be able to see Web3D's display without plug-ins [44].

The base format selection then switches to X3DOM which is published by a member of the Web3D Consortium, Fraunhofer [45]. Although not a standard format, X3DOM that connects HTML5 and X3D using DOM [46] brings several characteristics that are expected to avoid problems which can arise when using X3D. X3DOM makes it possible to combine Web3D's world view into conventional web pages so that it makes it more dynamic and interactive [47]. X3DOM is also one of the implementations of the need to shift towards declarative ways of describing 3D objects on the web [48], which makes the creation of 3D content on the web easier when compared to imperative methods. Figure 3 shows the X3DOM architecture which includes users of the web browser and X3D runtime.

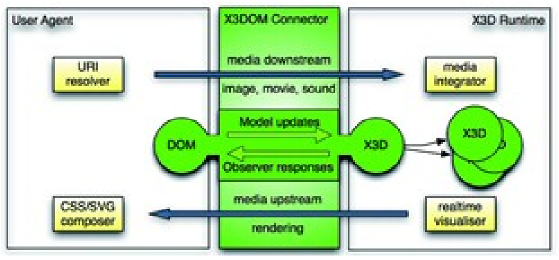


Figure 3: X3DOM Architecture [49]

5. RESOURCE MANAGEMENT

The initial part of the framework, the resource management, gathered the files obtained from the source sites and store it locally in certain folders. The entire system does not convert source world files in real time since there were difficulties found in the attempts to retrieve files from servers which host the world files. Therefore, direct conversion to produce results which can be displayed immediately in the browser can not be implemented. Some sites even block access to the world document file folder so that file retrieval must be done by manually checking the main file, and then manually save each incoming file.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Certain browsers allow access to world files that have been cached, even though the folder structure still has to be manually created to follow the source world original structure. Therefore, the main purpose of the first part of the framework was not about the collection process from the source site, but the management of the files that have been collected from various source sites. Figure 4 shows the overall fundamental process in the file management section of the framework.

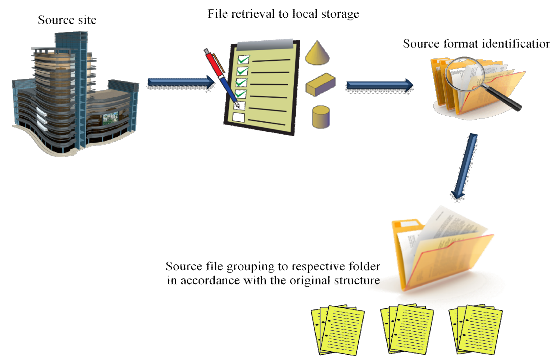

Source site
File retrieval to local storage
Source format identificatior
Source file grouping to respective folder in accordance with the original structure

Figure 4: Overall Resource Management Process

This section works by grouping the results of collecting files from the source site into folders that were specifically prepared to store the source world files. Grouping was done to ease the process and to avoid errors when accessing the file sources. For each group of source worlds to be converted, it was placed in a folder named after the title of the integration project to be carried out. The source world list and the title of the integration project were stored for easy identification. The structure of each folder from the source world was adapted to the folder structure of the original site so that the world can still be displayed in the offline browser, by including all the accompanying files especially the texture image file.

Not all source worlds can be used directly in conversion, because both VRML and X3D files can be compressed using gzip [50]. Especially for X3D, compression can also be done by converting text files into binary files using Compressed Binary Encoding/CBE techniques [51]. Compressed files will block the conversion process, thus requiring special treatment. The following is the processing method used to handle the overall source files, including the compressed file:

(1) make a list of contents using source world names along with all related information;
(2) each source world files that has been retrieved from the original site was placed in a certain source world folder, in a subfolder named with the number/name of the integration project, and in the sub-subfolder named the world specific identifier;
(3) set the naming of all sub-folders inside it to match the description of the source world, and also check the document content for each file to see the arrangement and name of the folder;
(4) check file types to find out the format used;
(5) prepare a batch command with the list of folders containing files that need to be converted;
(6) if the source files were in VRML format and they were compressed, they were placed in a particular folder for compressed files;
(7) the same was done to files in compressed X3D format;
(8) prepare a batch command containing the list of compressed files;
(9) run the decompression process on all compressed files;
(10) send the decompression results back to the source world folder, then create a batch contains a list of files to convert;
(11) register the types of formats from each source world entered in the project so that a specific command can be issued for a particular source world; and
(12) register the main files of each source world that will be used in the overall process as part of an integration project.

Figure 5 describes the process sequence in the resource management section before the conversion process was carried out. This activity assumes all files have been retrieved from the source world site complete with all components of the forming object, whether they were compressed or not. It is also assumed that all supporting files (image, sound, video, etc.) have also been localized first, even though these type of files do not get specific treatment other than just being copied to the next folder.

The process of checking and arranging object files follows the path that starts from the source world main file. Figure 6 shows a sample snippet of the source world document with its adjustments to the path arrangement in the source world file storage folder. The preparation of this folder assumes that the source world itself was not a distributed world, which means that if the source world refers to any external components or world, the referenced object or world will not be stored locally. Manually changing the source world external reference is needed so that the external resources can be assumed as local. The actual components or world also need to be localized and become part of the assets of the particular source world.

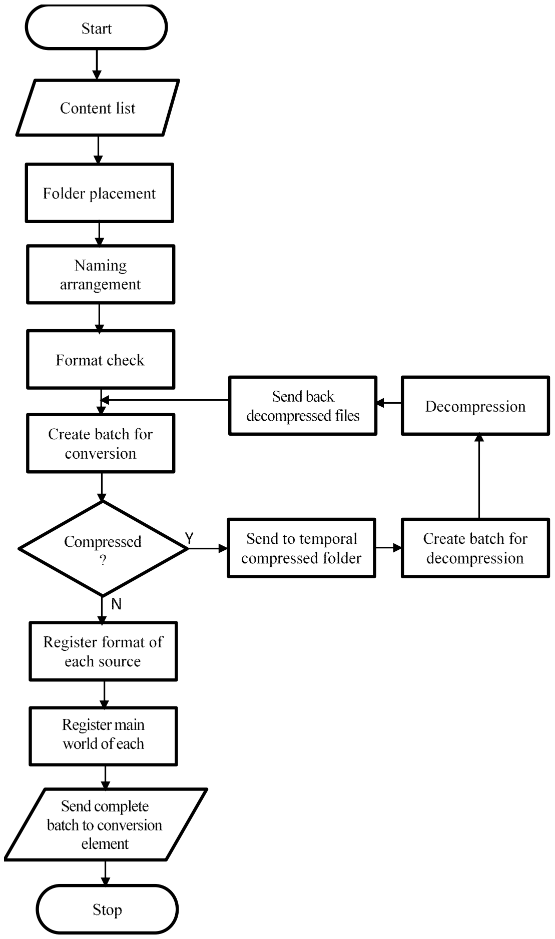

Start
Content list
Folder placement
Naming arrangement
Format check
Send back decompressed files
Decompression
Create batch for conversion
Compressed ?
Y
Send to temporal compressed folder
Create batch for decompression
N
Register format of each source
Register main world of each
Send complete batch to conversion element
Stop

Figure 5: Resource Management Process

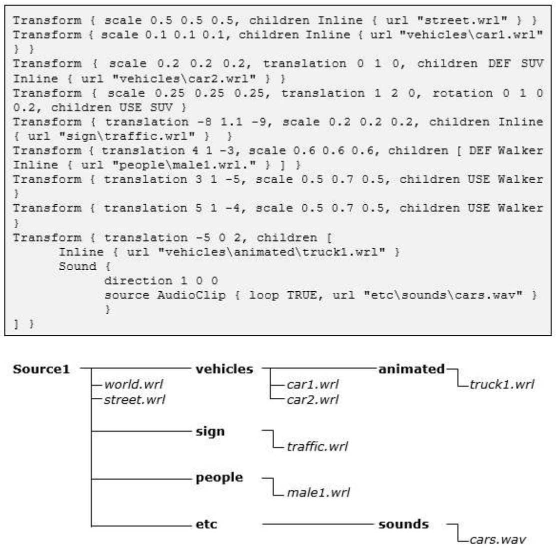

```
Transform { scale 0.5 0.5 0.5, children Inline { url "street.wrl" } }
Transform { scale 0.1 0.1 0.1, children Inline { url "vehicles\car1.wrl"
} }
Transform { scale 0.2 0.2 0.2, translation 0 1 0, children DEF SUV
Inline { url "vehicles\car2.wrl" } }
Transform { scale 0.25 0.25 0.25, translation 1 2 0, rotation 0 1 0
0.2, children USE SUV }
Transform { translation -8 1.1 -9, scale 0.2 0.2 0.2, children Inline
{ url "sign\traffic.wrl" }  }
Transform { translation 4 1 -3, scale 0.6 0.6 0.6, children [ DEF Walker
Inline { url "people\male1.wrl." } ] }
Transform { translation 3 1 -5, scale 0.5 0.7 0.5, children USE Walker
}
Transform { translation 5 1 -4, scale 0.5 0.7 0.5, children USE Walker
}
Transform { translation -5 0 2, children [
      Inline { url "vehicles\animated\truck1.wrl" }
      Sound {
            direction 1 0 0
            source AudioClip { loop TRUE, url "etc\sounds\cars.wav" }
            }
] }
```

Figure 6: Sample Storage Adjustments Based on Source World Document

6. CONVERSION

Results from the resource management were files that have been stored locally according to their original structure and set to be ready for the next section: format conversion. In general, the process in the conversion section in the framework is as shown in Figure 7. This section uses the source files that have been prepared in the local storage area, thus the entirety of conversion process was also done locally.

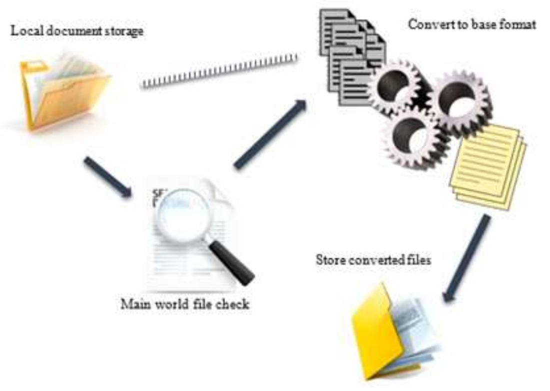


Figure 7: Overall Conversion Process

The basic model of the conversion system used in accordance with the proposed framework is to do the conversion locally on the user system. To achieve this goal, the framework does not rely on tools that are online via the internet, but rather use locally installed tools as part of the framework. As the framework is not a web-based application, the conversion system does not need to send files to certain servers on the internet and then retrieve the conversion results. Online method of conversion will take a longer time since there are additional steps to be done, and requires connectivity to the internet to use the framework. Aside from not being online, conversion systems also do not use web servers locally. The framework is a system that includes the tools needed and works like conventional computer applications. Conversion will be executed only when the batch contains a list of files from the source world has been sent by the resource management section.

Conversion was done to actual document contents from source worlds and not based on metadata. The basic principle of conversion used in the framework is to reduce user workload as much as possible with automation. Thus the user only carries out some important steps needed for the

conversion process within the framework to work. Activities that need to be carried out by users were:
(1) initiating a whole or certain part of the framework;
(2) choose the file to be converted; and
(3) inform the source format to the system.

There are various Web3D document formats currently available, some of which are mentioned in [52], [53], and [54], and each format has various features and specifications. Building a framework that covers all of the current Web3D formats will have to consider the availability of tools and libraries. Therefore, this study has limited the format that can be converted by the framework. Research on this framework was using VRML and X3D format issued by Web3D Consortium, as well as the X3DOM format issued by Fraunhofer IGD. These three formats act as source documents in the conversion process, and the results from the overall conversion was in X3DOM format.

Simply stated, the conversion process can be described in Figure 8 which shows the sequence after the file was retrived from the processed resource storage. The process starts with checking the source format, then the treatment was done after checking the format. If the source uses VRML format then it was converted to X3D, if the source was X3D, it will be copied directly to the particular converted folder. If the source file uses the X3DOM format, the X3D component will be taken and formed into the main file with the X3D format. The support files in X3D format will be copied directly to the converted folder. Basically the conversion process has reach its end when the result of conversion process has been stored locally, after which an update was made to the project information which shows the list of converted source world files and their structure.

Based on the process model, a series of X3D file preparation processes from the source file for all formats accommodated in the framework are prepared. Basic activities in the method used to convert the source files in VRML format were as following:
(1) open source file;
(2) preparing the destination file;
(3) initiation of readers and coding of new formats;
(4) write down the header of the destination file;
(5) read each source document and write new document in the destination file;
(6) performing data readings that accompany each code and writing data in the form of a destination format; and
(7) end writing by closing the source and destination files.

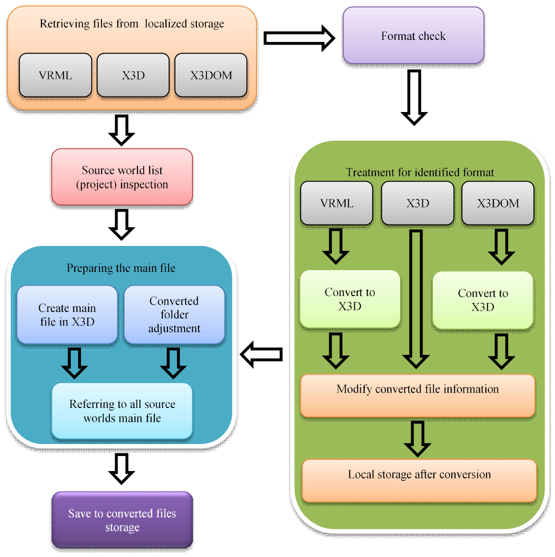

Retrieving files from localized storage
VRML
X3D
X3DOM
Format check
Source world list (project) inspection
Preparing the main file
Create main file in X3D
Converted folder adjustment
Referring to all source worlds main file
Save to converted files storage
Treatment for identified format
VRML
X3D
X3DOM
Convert to X3D
Convert to X3D
Modify converted file information
Local storage after conversion

Figure 8: Conversion Process Model

Once the source world was known to use the VRML format, this type of format was registered to the system along with the storage folder information in the active project. The use of the VRML format can be known from the extension used, whether it is .wrl, .wrz, or wrl.gz. A valid VRML document file starts with a header as seen in Figure 9.

```
#VRML V2.0 utf8
```

Figure 9: VRML Header

The header above is used for any VRML97 and VRML 2.0 documents, while VRML 1.0 used the one as seen in Figure 10.

```
#VRML V1.0 ascii
```

Figure 10: VRML 1.0 Header

As the use of VRML 1.0 was very rare, the one used in this framework was source files with VRML 2.0 or VRML97 format. Whereas the X3D coding used as the destination after conversion was XML coding, not classic VRML coding. The use of XML coding would provide convenience if there was any need to convert it to X3DOM. Using XML coding, the X3D document began with the following three lines as seen in Figure 11.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE     X3D     PUBLIC     "ISO//Web3D//DTD     X3D     3.0//EN"
"http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D            xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
profile='Full'                                          version='3.0'
xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-
3.0.xsd'>
```

Figure 11: X3D Header

Changing from VRML to X3D requires a lot of adjustments, because the writing pattern and node descriptions were different, especially because X3D uses XML style while VRML which derived from OpenInventor [55] has similarities with C language. However, because X3D is a direct superset of VRML [56] and its implementation [57], the information on object formation from documents with VRML format can still be used in X3D format. This causes the visual display of the converted object to look identical to its original format. A snippet from a sample file at the beginning of preparing an object in VRML format has been shown in Figure 12.

```
DEF jendelaluar23lt Transform {
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.5882 0.5882 0.5882
          ambientIntensity 1.0 specularColor 0 0 0 shininess 0.145
           transparency 0
        }
```

Figure 12: An Object Preparation in VRML

In X3D format, it was written as following as seen in Figure 13.

```
<Transform DEF='jendelaluar23lt'>
  <Shape>
    <Appearance>
      <Material  ambientIntensity='1'  diffuseColor='0.5882  0.5882
0.5882' shininess='0.145'/>
```

Figure 13: An Object Preparation in X3D

As seen in the X3D version of the document line, the values specified in VRML can be used in X3D with some adjustments in how to write them. Some attributes, specularColor and transparency are not written on lines that were converted from VRML, because the attribute value taken from VRML document was 0. The attribute with value 0 does not need to be written in the X3D document version since it is the default value.

Changes were also made to how to describe coordinates of an object. In VRML, each set of coordinates was written in the order x, y, and z followed by a comma, then writes the next coordinate set, as seen in Figure 14.

```
coord DEF jendelaluar23lt-COORD Coordinate { point [
        6 0 -125, -6 0 -125, -6 0 125, 6 0 125, 6 5 -125, -6 5 -
125,
```

Figure 14: Coordinate Description in VRML

Writing the coordinate set in the X3D format was basically the same because it uses the order of the x, y, and z values. The difference was only it doesn't use comma after each set of coordinates. Even so, the values used were the same. From the document snippet above, it was written in the X3D format as seen in Figure 15.

```
<Coordinate DEF='jendelaluar23lt-COORD' point='6 0 -125 -6 0 -125 -
6 0 125 6 0 125 6 5 -125 -6 5 -125
```

Figure 15: Coordinate Description in X3D

Similar changes were also made to other parts, one of which was in describing the position to place the image file that was made into a texture map to the object's surface. In VRML, it was written in the same way as the following document snippet shown in Figure 16.

```
texCoord DEF jendelaluar23lt-TEXCOORD TextureCoordinate { point [
     0.9918 0.2856, 0.007084 0.2856, 0.4929 0.2855, 0.5082 0.2856,
     0.9918 0.2936,
```

Figure 16: Texture Placement in VRML

The information was written in the X3D document as seen in Figure 17.

```
<TextureCoordinate   DEF='jendelaluar23lt-TEXCOORD'   point='0.9918
0.2856 0.007084 0.2856 0.4929 0.2855 0.5082 0.2856 0.9918 0.2936
```

Figure 17: Texture Placement in X3D

The same way is done on the coordinate index for textures, in VRML documents were written in the following lines shown in Figure 18.

```
texCoordIndex [
          973, 1, 5, -1, 5, 972, 973, -1, 1, 2, 6, -1, 6, 5, 1, -1,
          2, 3, 7, -1, 7, 6, 2, -1, 3, 0, 4, -1, 4, 7, 3, -1,
          973, 974, 975, -1, 975, 1, 973, -1, 977, 978, 979, -1,
```

Figure 18: Coordinate Index for Textures in VRML

Whereas in X3D it was written in the following way as seen in Figure 19.

```
texCoordIndex='973 1 5 -1 5 972 973 -1 1 2 6 -1 6 5 1 -1 2 3 7 -1 7
6 2 -1 3 0 4 -1 4 7 3 -1 973 974 975 -1 975 1 973 -1 977 978 979 -1
```

Figure 19: Coordinate Index for Textures in X3D

Nodes for other visual effects also get adjustments as needed, including normal vertices. Normal vertices were usually elaborated in non-primitive complex polygons because it was used to indicate perpendicular direction from the arrival of light, so that it can be used to manipulate the shade as a result of lighting which landed on the surface of the object. In VRML format, it can be written in the following sample document snippet as seen in Figure 20.

```
normal Normal { vector [
          -0.8778 0 0.4788, 0.8778 -0.4788 0, 0.8778 0 -0.4788,
          1 0 0, -0.8778 -0.4788 0, -0.8778 0 -0.4788, -1 0 0,
          0 1 0, 0 0 1, 0.8778 0.4788 0, 0 -1 0, 0.8778 0 0.4788,
          0 0 -1, -0.8778 0.4788 0, ] }
```

Figure 20: Normal Vertices in VRML

In X3D format, the value of the normal vertex was written as seen in Figure 21.

```
<Normal vector='-0.8778 0 0.4788 0.8778 -0.4788 0 0.8778 0 -0.4788
1 0 0 -0.8778 -0.4788 0 -0.8778 0 -0.4788 -1 0 0 0 1 0 0 0 1 0.8778
0.4788 0 0 -1 0 0.8778 0 0.4788 0 0 -1 -0.8778 0.4788 0'/>
```

Figure 21: Normal Vertices in X3D

The next translations were on the coordinate index. In documents that describe complex objects, this section was used to define coordinates used to draw the surface of an object. In VRML, this coordinate index was written in a way like the following lines of document shown in Figure 22.

```
coordIndex [
          0, 1, 5, -1, 5, 4, 0, -1, 1, 2, 6, -1, 6, 5, 1, -1, 2, 3,
```

Figure 22: Coordinate Index in VRML

Whereas in X3D, lines to define the index coordinates were written as in the following line of document lines shown in Figure 23.

```
coordIndex='0 1 5 -1 5 4 0 -1 1 2 6 -1 6 5 1 -1 2 3
```

Figure 23: Coordinate Index in X3D

Conversions were also made on document files from source worlds with X3DOM format. Since the world with the X3DOM format only uses this format for the main world file, then the conversion was only done on the main world file. In the case of converting X3DOM files to X3D, the steps that have been done were: (1) removing HTML elements and (2) making adjustments to the lines of documents from X3DOM to meet the X3D writing style. For example, based on the same object used in sample conversion from VRML to X3D, if the object was written in X3DOM format then the initial section of the document contains the following lines as seen in Figure 24.

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv='Content-Type' content='text/html;charset=utf-
8'></meta>
    <link              rel='stylesheet'              type='text/css'
href='http://www.x3dom.org/x3dom/release/x3dom.css'></link>
    <script                                   type='text/javascript'
src='http://www.x3dom.org/x3dom/release/x3dom.js'></script>
  </head>
  <body   style='width:100%;   height:100%;   border:0;   margin:0;
padding:0;'>
    <x3d id='jendaluar23' showStat='false' showLog='false' x='0px'
y='0px'   style='width:100%;   height:100%;   border:0;   margin:0;
padding:0;'>
      <scene DEF='scene'>
        <transform DEF='jendelaluar23lt'>
          <shape>
            <appearance>
              <material  ambientIntensity='1'  diffuseColor='0.5882
0.5882 0.5882' shininess='0.145'></material>
```

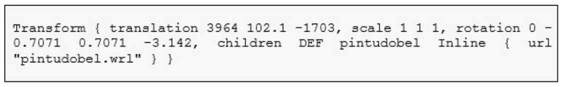
Figure 24: Sample Document in X3DOM

In accordance with the specifications in X3DOM, the first line of document identifier indicates that the file is an HTML5 document. The next HTML element up to the <x3d> element is part of the HTML that makes the X3D file X3DOM. To convert it to an X3D document, the HTML element was replaced with an X3D header as mentioned in the previous section. Since X3DOM uses the method of writing attributes as in HTML documents, the attributes in X3DOM are also not case sensitive [58], including all X3D elements specified in the document. When converted to X3D, these elements were rewritten according to the syntax in the X3D rules, including following the terms of use of upper and lower case letters in the document. The following were lines of document based on the previous sample written in X3D format, shown in Figure 25.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE     X3D     PUBLIC     "ISO//Web3D//DTD     X3D     3.0//EN"
"http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D            xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance'
profile='Full'                                           version='3.0'
xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-
3.0.xsd'>
    <Scene DEF='scene'>
    <Transform DEF='jendelaluar23lt'>
      <Shape>
        <Appearance>
          <Material  ambientIntensity='1'  diffuseColor='0.5882  0.5882
0.5882' shininess='0.145'/>
```

Figure 25: Sample Document in X3D

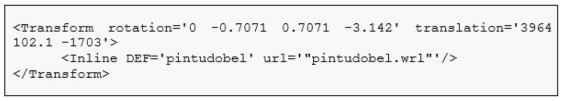
Changes made in the context of conversion were carried out on all source files with VRML and X3DOM formats. The conversion results were stored in the folder prepared for files that have been converted into X3D format. After all files were uniformed in X3D format, the next step was to correct the reference to the external file contained in the converted documents. Reference to external resource in a VRML file can be seen in Figure 26.



```
Transform { translation 3964 102.1 -1703, scale 1 1 1, rotation 0 -
0.7071  0.7071  -3.142,  children  DEF  pintudobel  Inline  {  url
"pintudobel.wrl" } }
```

Figure 26: Reference to External Resource in A VRML File

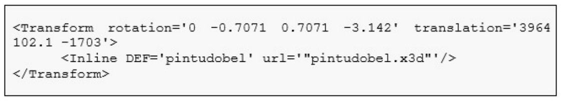
The lines were converted to be like the one shown in Figure 27.



```
<Transform  rotation='0  -0.7071  0.7071  -3.142'  translation='3964
102.1 -1703'>
      <Inline DEF='pintudobel' url='"pintudobel.wrl"'/>
</Transform>
```

Figure 27: Reference to External Resource in An X3D File

In the X3D documents written based on the original VRML document, there was a reference to a file in VRML format. This would obviously cause errors because all files were uniformed to X3D format. Therefore, file reference must also be adjusted to the correct file resulting from the conversion. The line of documents were changed to the following as seen in Figure 28.



```
<Transform  rotation='0  -0.7071  0.7071  -3.142'  translation='3964
102.1 -1703'>
      <Inline DEF='pintudobel' url='"pintudobel.x3d"'/>
</Transform>
```

Figure 28: Reference to External Resource in An X3D File, Corrected

As the scale contains no modification in object size, it was not written in X3D format. In addition, it was also necessary to adjust the folder name. This would only be done if there were any further change in project folder name since the list that the conversion section worked on contains all the folder name of the source world.

7. RESULTS AND DISCUSSION

Resource management section that work at the beginning of the framework were tasked with providing the data needed so that the next element can work according to its purpose. The main purpose of data management was to ensure that the source world folder structure was used in accordance with the original. Resource management was needed because when the collection of files from the source site was done, often the source world files accumulate into one folder, or spread to various folders created during retrieval and did not match the original folder structure. This buildup can occur because sites that function as hosts for the source world do not provide direct access to the folder containing the required files. If something like this happens, active retrieval cannot be done to the server on that site.

Tools for retrieving files, especially those in the mass-downloader category, will also not be used if access to the folder is blocked. To overcome this, the required files were obtained through passive retrieval, by retrieving the files obtained from the server when browsing in the world. Passive retrieval was done in two ways: (1) searching and retrieving source world files from the cache folder in the browser used to browse the source site; or (2) calling each component of the source world one by one using a browser.

Passive retrieval through the cache actually makes it easy to collect files from the source site, because if browsing was done completely by passing and interacting with all the elements in the source world Web3D site, all the required files will be available in the cache. In some browsers, file names that are components in the source world site were shown as they were on the original server, whereas in certain browsers the file name was made random, making it difficult to search for the required files and requiring additional time to change the file name to the original. However, even though all files can be available in the cache and can be identified, all the files were stacked in just one folder or scattered in several browser cache folders according to the session. Thus, it is necessary to rebuild the placement of all files that have been obtained in order to have the folder structure just as the original. Resource management section carried out this task so that world resources can be displayed offline on local computers without being connected to the internet, which means they were ready to be converted by the next framework section.

Passive retrieval by calling each component based on the reference that starts from the main file of the world can become a significantly time consuming process if there were many different components of the forming component, and there were sub-references to the components taken. If there were many references to similar components, then passive retrieval in this way would not require very much effort, because the component files taken were not too many in numbers. The time required can also be shorter when the referred component does not refer to other smaller forming components. In general, this method takes longer when compared to using a cache, but this method can better ensure the availability of components from the source world. This method also makes it possible to directly get the structure that matches the folder structure on the source site server if the file obtained was stored directly in the path sequence that matches the description in the file that refers to the particular component. If the source world files were obtained in this way, the task of the file management element was to match the structure that has been obtained from the collection of files with the path mentioned in the source world files, especially from the main file.

Another task done by the file manager was to decompress the source world file. Without decompression, the conversion will not be possible because the compressed file cannot be read. This element checks the file extension to determine whether the files need to be converted or not. Type checking were carried out in the following ways: (1) in the source world with VRML format, the decompression command will be given when the .wrz or .wrl.gz extension were found in files stored in folders and all subfolders. The problem that can arise when using this method was that VRML allows the use of .wrl extension on files that have been compressed. This allows errors in the execution of the next section because the compressed file was considered uncompressed and the conversion attempt will be carried out immediately.
(2) Whereas in the source world with the X3D format, the decompression command will be given when any files with the extension .x3dz, x3d.gz, .x3dvz, .x3dv.gz, .x3db, or .x3db.gz in the folder and all subfolders were found. Files with .xd3b extension was among those that must be decompressed because this file was written with the CBE method which made the file contents became a set of binary data. In order to gain access to the ASCII form of this binary file, decompression was performed. The result was a file in X3D format so it was not included in the conversion treatment.

The most important task of the resource management element was to prepare a list of files to be converted by the conversion section. Preparation of a file list was needed because the framework was prepared to minimize user's work, by reducing the workload of changing the format, especially if the files which needed to be converted were in large numbers. File management section created a list of files that needed to be converted to all folders in each source world, then send a list of files according to the source world folder structure to the conversion section. The list of files was obtained through searching each folder for a particular file type. The process were done sequentially by the system so that no source format file was missed.

This framework section was tested in an experiment to manage three sample source worlds in VRML, X3D, and X3DOM formats that were localized in a folder that was assumed to be collected from the source site. Some of the files in each source world for experiments contain compressed files, all files have not been arranged in the folder structure according to the path description in the source world, and the list of files ready for conversion was not available. The resource management section begins by registering the source world into an integration project accompanied by mentioning the initial location of folders used to store the source world files respectively. The management process ends after the project folder contains all source world folders have been prepared, and a list of all files to be converted has been completed. Table 1 shows the results of this section test.

Table 1. Results from file management test

Source/ format	3D files/ compressed	Other files	Folder placement	File decompression	File list
World 1/ VRML	54/44	14	100%	73%	100%
World 2/ X3D	73/21	11	100%	100%	100%
World 3/ X3DOM	66/38	23	100%	37%	100%

The resource management section implementation has able to prepare the file list and set the location of the file to be converted. However, the decompression sub-system still encountered obstacles since the VRML file management has not been able to distinguish between compressed or uncompressed files when the VRML files were using the same extension (.wrl). There were 12 compressed files in .wrl extension that were not decompressed. Whereas in the X3DOM file, the resource management section could not process 24 binary compressed files that use the BinaryGeometry node from the InstantReality platform [59]. The X3DOM file used in the experiment can refer to X3D files that were compressed using gzip in both XML encoding [60] and classic VRML [61], but could not refer to CBE binary compressed files which were X3D standards. In order to have the minimized file size feature as well as 3D structure optimization, source world in X3DOM format can utilize BinaryGeometry which is not a standard node in X3D.

The next part was the conversion section. This section was simply a part that produces the files needed to be displayed in the browser. As the second part of the framework after the resource management section, the conversion section received a list of conversion target files from each source world. The first step that needed to be done was to check the format information of the file to be converted. This was needed so that the conversion section can properly convert the files to the destination format. As per the initial design, if the source file format was X3D then no conversion will be made, thus the conversion element will only prepare the conversion process to files with formats other than X3D.

The next step was inserting the command to read the source file and create the target file in a new format, for each file listed in the conversion list. Based on each command on the file that was included in the list for a world, the conversion process was carried out. The process of reading the source file from the folder that has been prepared in the resource management process and writing new files in the conversion folder was repeated on all files in the list of each source world. The repeated process was including copying any X3D files to the particular converted folder. Each converted or copied file was placed in the same structure as set in the folder of each source world.

The process in the conversion section did not change the conversion result back into compressed form. If this was done, then the adjustment process will experience difficulties if the source world has too many differences in display properties compared to other source worlds. In order to avoid errors in the test, all previous files in VRML format and the ones failed to be decompressed in the resource management section needed to be decompressed first. Whereas binary files from source world in X3DOM format were treated like X3D files that will be referred to by the primary file in X3DOM format, therefore these files were not converted.

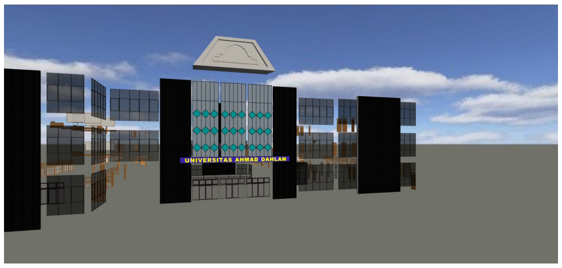
Based on the conversion process for each source world file type, the results obtained can be shown in the attached Table 2. All variants of the source format sent by the resource management section have been converted in this section, except for files that were not prepared as input for the conversion system. It can be concluded that all source world conversion results can perform well in the browser just like when they were displayed in their original format. As for the entire source world, the process results from the conversion section based on the tests that have been carried out can be seen in the attached Table 3.

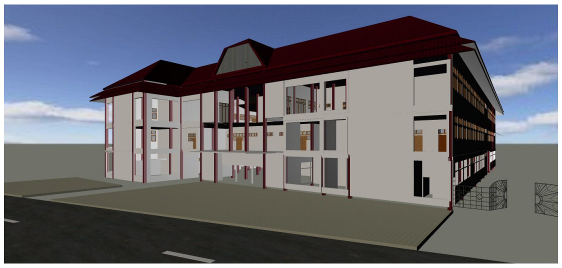
The process in the conversion section has placed all the conversion results as well as copies of files from source world to the designated converted folder and in accordance with the original folder structure. As per the initial design, all files that have used the X3D format were only be copied to the converted folder without any further changes. All files from the VRML format have been converted, and only one file from the X3DOM format was converted because the other files of the particular world have used the X3D format and some were in binary form using BinaryGeometry node. Deformation of 3D objects did not occur in the world as the result of conversion, but there was a slight discrepancy in mapping the textures to objects that used BinaryGeometry nodes.

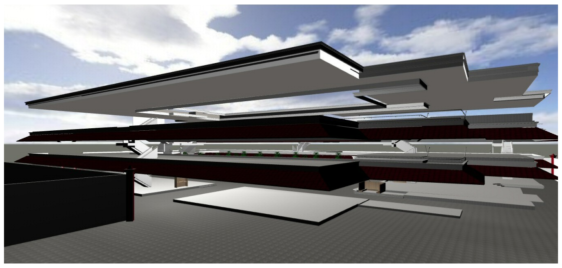
Overall sequence of processes and their related data were shown in attached Figure 29. Except for the planned adjustment process which was not yet included in this research, the framework has able to carry out all the specific process, leaving only the adjustments which needed to be done manually as the particular element has not yet prepared.

Source worlds with their respective contents in VRML, X3D and X3DOM formats were shown in Figure 30. Specificity can be seen in X3DOM world which used X3D documents as its components. Since the X3DOM world also used BinaryGeometry binary files, they were placed in the binGeo folder copied from the source world. Each source world contains various objects with a ratio of sizes that have been uniformed, the same overall direction when viewed from the same camera point, and the same initial coordinate points or object location. Manual equalization to the position, size, and direction attributes was done before all processes in the framework were carried out because the framework discussed in this paper was only consisted of the resource management and conversion sections. The framework that has been completed in this research did not cover the adjustment section and the results preparation. Source worlds that have the right direction, location, and scale can be directly integrated with each other to form a single world that seems proper and can be directly explored.

Using the implementation which consisted of two parts of the framework, the source worlds were processed and the results were shown in the browser, as shown in Figure 31. The final world has used the X3DOM format as the base format as planned in the overall framework. In the initial design, the X3DOM format will be used at the last element which has not been included in this first version of the framework. However, the concept of using primary files in X3DOM format as the main world that act as the caller of all conversion results has been used up to this section and implemented manually through the use of the primary world file template. The aim was that the framework could accommodate files that used the BinaryGeometry node since this file was not among the converted files.


(a)


(b)


(c)
Figure 30: Source worlds: (a) in VRML; (b) in X3D; (c) in X3DOM

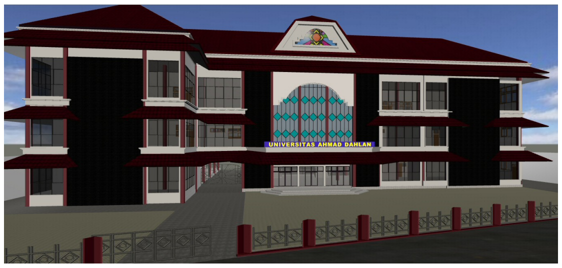


Figure 31: Integrated worlds

8. CONCLUSION AND FUTURE WORKS

The multi-format contribution for the distributed world principle has been implemented into a working framework. The framework that included two main components, the resource management section and the conversion section, has been able to provide a series of processes which were needed to integrate the separated world into a single world. The framework has solved the problem of differences in formats of the source worlds by accommodating a limited number of formats, although it has not been prepared to handle differences in views based on the size, position, and direction of the content contained in each world. The final world produced by the framework allows seamless integration of previously separate worlds, so that users can explore all of the world as if they navigate and interact in a single, very large world.

As part of an overall full framework, the research only covered the first two parts of all four elements of the planned framework. Subsequent research will complement this framework so that it can present a complete concept of a framework that will enable users to interact with several worlds at the same time without having to switch browsers, with little performance degradation on their computer systems. Future research will also add more format that can be covered by the framework, to expand the range of world diversity that can be integrated.

REFERENCES:

[1] Don Brutzman and Leonard Daly, "Extensible 3D Graphics for Web Authors", Morgan Kauffman Publishers, San Fransisco, 2007.

[2] Arne Schilling and Alexander Zipf, "Generation of VRML city models for focus based tour animations: integration, modeling and presentation of heterogeneous geo-data sources", *Proceedings of the Eighth International Conference on 3D Web technology (Web3D '03)*, Saint Malo (France), March 09 - 12, 2003, pp. 39-ff.

[3] Sanjin Jeginovic, "Interactive 3D models – From 3ds max to VRML", *Proceedings of 8th Central European Seminar on Computer Graphics (CESCG)*, Budmerice Castle, 2004.

[4] Masood Masoodian, Azmi Bin Mohd Yusof, Bill Rogers, "Supporting Focus and Context Awareness in 3D Modelling Tasks Using Multi-Layered Displays", *Computer Graphics Forum*, Volume 34, Issue 6, Sept 2015. pp. 1–12.

[5] Web3D Consortium, "VRML Content Authoring & Editing Tools", *https://www.web3d.org/x3d/vrml/tools/authoring/*, February 28, 2005.

[6] Web3D Consortium, "X3D Resources", *http://www.web3d.org/x3d/content/examples/X3dResources.html#AuthoringTools*, 2018.

[7] Javier Barbadillo and Jairo R. Sánchez, “A Web3D authoring tool for augmented reality mobile applications”, *Proceedings of the 18th International Conference on 3D Web Technology (Web3D '13)*, San Sebastian, Spain, 2013, pp. 206-206.

[8] Cortona3D, “RapidAuthor”, *http://cortona3d.com/rapidauthor*, 2018.

[9] Bitmanagement.com, “BS SDK”, *http://www.bitmanagement.com/en/products/authoring-tools/bs-sdk*, 2018.

[10] Kostas Kapetanakis and Spyros Panagiotakis, “Evaluation of Techniques for web 3D Graphics Animation on Portable Devices”, *Proceeding of International Conference on Telecommunications and Multimedia (TEMU)*, 30 July-1 August 2012, Chania, Greece, pp. 152-157.

[11] Rui Yang, WenJie Fan, and Jingrong Sun, “Application in the Training of Automatic Weather Station Based on Web3D Technology”, *International Journal of Emerging Technologies in Learning*, Vol 10, No 4, 2015, pp. 34-39.

[12] Nathan A. Curtis, “Modular Web Design: Creating Reusable Components for User Experience Design and Documentation”, New Riders, Berkeley, CA, 2010.

[13] Mark Pesce, “VRML: Browsing & Building Cyberspace”, New Riders Publishing, Berkeley, CA, 1995.

[14] Alun Evans, Marco Romeo, Arash Bahrehmand, Javi Agenjo, and Josep Blat, “3D graphics on the web: A survey”, *Computers & Graphics*, Volume 41, 2014, Pages 43-61.

[15] Zhang Huimin, Chen Manqing and Chen Zhaojun, "Based on VRML online virtual campus design", *Proceedings of International Conference on Computer Application and System Modeling (ICCASM 2010)*, Taiyuan, 2010, pp. V14-398-V14-400.

[16] Li Ping Liu and Yu Jian Wang, "An Implement of Virtual Campus System Base on the X3D Technology", *Advanced Materials Research*, Vol. 422, 2012, pp. 551-554.

[17] Raj Kishen Moloo, Sameerchand Pudaruth, Mahendranath Ramodhin and Reezwanah Bibi Rozbully, "A 3D Virtual Tour of the University of Mauritius using WebGL", *Proceedings of International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, 2016, pp. 2891-2894.

[18] Web3D Consortium, “Recommended Standards”, http://www.web3d.org/standards, 2018.

[19] International Organization for Standardization, “ISO/IEC 14772-2:2004 - Information technology -- Computer graphics and image processing -- The Virtual Reality Modeling Language (VRML) -- Part 2: External authoring interface (EAI)”, *https://www.iso.org/standard/30893.html*, 2004.

[20] International Organization for Standardization, “ISO/IEC 19775-1:2013 - Information technology -- Computer graphics, image processing and environmental data representation -- Extensible 3D (X3D) -- Part 1: Architecture and base components”, *https://www.iso.org/standard/60760.html*, 2013.

[21] Aaron E. Walsh, and Mikaël Bourges-Sévenier, “Core Web3D”, Prentice Hall PTR, Upper Saddle River, NJ, 2001.

[22] Bartosz Sawicki, and Bartosz Chaber, “Efficient visualization of 3D models by web browser”, *Computing*, Volume 95 (Supplement 1), 2013, pp. 661-673.

[23] Daniel Limberger, Marcel Pursche, Jan Klimke, and Jürgen Döllner, “Progressive high-quality rendering for interactive information cartography using WebGL”, *Proceedings of the 22nd International Conference on 3D Web Technology (Web3D '17)*, Brisbane, Queensland, Australia, Article no. 8, June 05 - 07, 2017, 4 pages.

[24] Jacek Jankowski and Stefan Decker, “A dual-mode user interface for accessing 3D content on the world wide web”, *Proceedings of the 21st international conference on World Wide Web (WWW '12)*, Lyon, France, April 16 - 20, 2012, pp. 1047-1056.

[25] Dante Abate, Raffaele Ciavarella, Graziano Furini, Guido Guarnieri, Silvio Migliori, and Samuele Pierattini, “3D modeling and remote rendering technique of a high definition cultural heritage artefact”, *Procedia Computer Science*, Volume 3, 2011, pp. 848-852.

[26] Jussara M. Kofuji, Sergio T. Kofuji, and Marcelo Zuffo, “Analysis of X3D scene, Web3D objects and media panoramas”, *Proceedings of the 2010 Summer Computer Simulation Conference (SCSC '10)*, Ottawa, Ontario, Canada, July 11 - 14, 2010, pp. 40-47.

[27] Rozenn Bouville Berthelot, Jérôme Royan, Thierry Duval, and Bruno Arnaldi, “Scene graph adapter: an efficient architecture to improve interoperability between 3D formats

and 3D applications engines", *Proceedings of the 16th International Conference on 3D Web Technology (Web3D '11)*, Paris, France, June 20 - 22, 2011, pp. 21-29.

[28] Rozenn Bouville Berthelot, Jérôme Royan, Thierry Duval, and Bruno Arnaldi, "3DFC: a new Container model for 3D File formats compositing", *Proceedings of the 17th International Conference on 3D Web Technology (Web3D '12)*, Los Angeles, California, August 04 - 05, 2012, pp. 27-35.

[29] Jean-Francis Balaguer and Enrico Gobbetti, "i3D: a high-speed 3D Web browser", *Proceedings of the first symposium on Virtual reality modeling language (VRML '95)*, San Diego, California, USA, December 13 - 15, 1995, pp. 69-76.

[30] Bitmanagement.com, "BS Contact", *http://www.bitmanagement.com/en/products/interactive-3d-clients/bs-contact*, 2017.

[31] Kenton McHenry, Rob Kooper and Peter Bajcsy, "Towards a Universal, Quantifiable, and Scalable File Format Converter," *Proceedings of Fifth IEEE International Conference on e-Science*, Oxford, 2009, pp. 140-147.

[32] Peter Bajcsy, Rob Kooper, Luigi Marini, Kenton McHenry, and Michal Ondrejcek, "A framework for understanding file format conversions", *Proceedings of the 2010 Roadmap for Digital Preservation Interoperability Framework Workshop (US-DPIF '10)*, Gaithersburg, Maryland, USA, March 29 - 31, 2010, Article 10, 7 pages.

[33] Kenton McHenry, Michal Ondrejcek, Luigi Marini, Rob Kooper, and Peter Bajcsy, "Towards a Universal Viewer for Digital Content", *Procedia Computer Science*, Volume 4, 2011, pp. 732-739.

[34] Mursid 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.

[35] Mursid W. Hananto, Ahmad Ashari and Khabib Mustofa, "Comparative study of Web3D standard format to determine the base format for a Web3D framework," *Proceedings of the 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Yogyakarta, 2017, pp. 1-6.

[36] Douglas Maxwell and Michael Heilmann, "Leveraging HTML5 and WebGL to Address Information Assurance Barriers for Simulation Based Training in the U.S. Military", *Proceedings of MODSIM World*, 25-27 April 2017, Virginia Beach, VA. Paper no 0001, 10 pages.

[37] Iñaki Prieto, Jose Luis Izkara, and Francisco Javier Delgado del Hoyo, "Efficient Visualization of the Geometric Information of CityGML: Application for the Documentation of Built Heritage", *Proceedings of Computational Science and Its Applications (ICCSA 2012)*, Springer, Berlin/Heidelberg, Germany, 2012; pp. 529–544.

[38] J. Guerrero, Sisi Zlatanova, and Martijn Meijers, "3D Visualisation of Underground Pipelines: Best Strategy For 3D Scene Creation", *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-2/W1, 2013, pp. 139-145.

[39] Bo Mao, "Visualisation and Generalisation of 3D City Models", *Doctoral Thesis*, Royal Institute of Technology (KTH), School of Architecture and the Built Environment (ABE), Urban Planning and Environment, Geoinformatik och Geodesi, Stockholm, 2011, p. xii, 104

[40] Iñaki Prieto and Jose Luis Izkara, "Visualization of 3D city models on mobile devices", *Proceedings of the 17th International Conference on 3D Web Technology*, Los Angeles, California, August 04 - 05, 2012, pp. 101-104.

[41] W. Zhang, H. Yuan, J. Wang and Z. Yan, "A WebGL-based method for visualization of intelligent grid", *Proceedings of the 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*, Weihai, Shandong, 2011, pp. 1546-1548.

[42] Tim Nicolas Eicke, Yvonne Jung, and Arjan Kuijper, "Stable dynamic webshadows in the X3DOM framework", *Expert Systems with Applications*, Volume 42, Issue 7, 2015, pp. 3585-3609.

[43] Haitao Wang, Xiaoyu Chen, Nicholas Polys, and Peter Sforza, "A Web3D forest geo-visualization and user interface evaluation", *Proceedings of the 22nd International Conference on 3D Web Technology* (Web3D '17), ACM, New York, NY, USA, Article 9, 2017, 9 pages.

[44] Fabio Pittarello, "Testing the X3DOM framework for the development of Web3D

applications", *Proceedings of the 18th International Conference on 3D Web Technology (Web3D '13)*, San Sebastian, Spain, June 20 - 22, 2013, pp. 191-194.

[45] Web3D Consortium, "HTML 3D", *http://www.web3d.org/html-3d*, 2018.

[46] Johannes Behr, Peter Eschler, Yvonne Jung, and Michael Zöllner, "X3DOM: a DOM-based HTML5/X3D integration model", *Proceedings of the 14th International Conference on 3D Web Technology (Web3D '09)*, Stephen N. Spencer (Ed.). Darmstadt, Germany, June 16 - 17, 2009, pp. 127-135.

[47] Johannes Behr, Yvonne Jung, Timm Drevensek, and Andreas Aderhold, "Dynamic and interactive aspects of X3DOM", *Proceedings of the 16th International Conference on 3D Web Technology (Web3D '11)*, Paris, France, June 20 - 22, 2011, pp. 81-87.

[48] Jean Le Feuvre, "Towards Declarative 3D in Web Architecture", *Proceedings of the 1st International Workshop on Declarative 3D for the Web Architecture (Dec3D2012 at WWW2012)*, Lyon, France, April 17, 2012.

[49] Johannes. Behr, Yvonne Jung, J. Keil, Timm Drevensek, M. Zoellner, Peter Eschler, and D. Fellner, "A scalable architecture for the HTML5/X3D integration model X3DOM", *Proceedings of the 15th International Conference on Web 3D Technology (Web3D '10)*, Los Angeles, California, July 24 - 25, 2010, pp. 185-194.

[50] Web3D Consortium, "X3D Binary Compression Capabilities and Plans", *http://www.web3d.org/wiki/index.php/X3D_Binary_Compression_Capabilities_and_Plans*, 2017.

[51] Web3D Consortium, "Extensible 3D (X3D) encodings, ISO/IEC 19776-3.2:2011, Part 3: Compressed binary encoding", *http://www.web3d.org/documents/specifications/19776-3/V3.2/Part03/X3D_Binary.html*, 2011.

[52] Kenton McHenry, and Peter Bajcsy, "An Overview of 3D Data Content, File Formats and Viewers, Technical Report: isda08-002", *Image Spatial Data Analysis Group*, NCSA, Urbana-Champaign, IL, 2008.

[53] Beata Turonova, "3D Web Technologies And Their Usability for The Project 3D Mobile Internet, Technical Report RDC-TR-09-8", *Research and Development Center for Mobile Applications*, Faculty of Electrical Engineering, Czech Technical University in Prague, 2009.

[54] Mikael Waerner, "3D Graphics Technologies for Web Applications, An Evaluation from the Perspective of a Real World Application", *Master Thesis*, Department of Electrical Engineering, Information Coding, Linköping University, The Institute of Technology. Linköping, 2012.

[55] Jeremy Walton, "Publishing in a 3D World: Data Sharing with VRML", *IRIS Explorer Technical Report*, TR7/96 IETR/5, NAG Ltd, 1996.

[56] Web3D Consortium, "What is X3D?", *http://www.web3d.org/x3d/content/examples/X3dResources.html*, 2018.

[57] David Arendash, "The unreal editor as a Web 3D authoring environment", *Proceedings of Ninth International Conference on 3D Web Technology*, Monterey, California, USA, 2004, pp.119-126.

[58] Fraunhofer IGD, "Official X3DOM Documentation: Introduction", *https://doc.x3dom.org/author/index.html*, 2018.

[59] InstantReality, "Download the instantreality Framework 2.8.0", *http://www.instantreality.org/downloads/*, 2016.

[60] Web3D Consortium, "Extensible 3D (X3D) encodings, Part 1: Extensible Markup Language (XML) encoding", *http://www.web3d.org/documents/specifications/19776-1/V3.2/Part01/concepts.html*, 2009.

[61] Web3D Consortium, "Extensible 3D (X3D) encodings, Part 2: Classic VRML encoding", *http://www.web3d.org/documents/specifications/19776-2/V3.2/Part02/concepts.html*, 2008.

Table 2. Converted Source Format

Format	Extension	# of files	Converted	Object placement	Invididual viewable	World viewable	Deformation	Texture/ coloring
VRML	.wrl	22	√	√	√	√	×	√
	.wrl.gz	24	√	√	√	√	×	√
	.wrz	8	√	√	√	√	×	√
X3D	.x3d	46	×	√	√	√	×	√
	.x3dv	4	×	√	√	√	×	√
	.x3dz	5	×	√	√	√	×	√
	.x3dvz	5	×	√	√	√	×	√
	.x3d.gz	3	×	√	√	√	×	√
	.x3dv.gz	5	×	√	√	√	×	√
	.x3db	2	×	√	√	√	×	√
	.x3db.gz	3	×	√	√	√	×	√
X3DOM	.html	1	√	√	√	√	×	√
	.x3d	20	×	√	√	√	×	√
	.x3dz	9	×	√	√	√	×	√
	.x3dv	7	×	√	√	√	×	√
	.x3dvz	5	×	√	√	√	×	√
	.bin+4	16	×	√	×	√	×	×
	.bin+8	8	×	√	×	√	×	×

Table 3. Results from File Conversion Test

Source world	Format	# of 3D files	Converted	Percentage	Folder placement
World 1	VRML	54	54	100%	100%
World 2	X3D	73	0	0%	100%
World 3	X3DOM	66	1	2%	100%

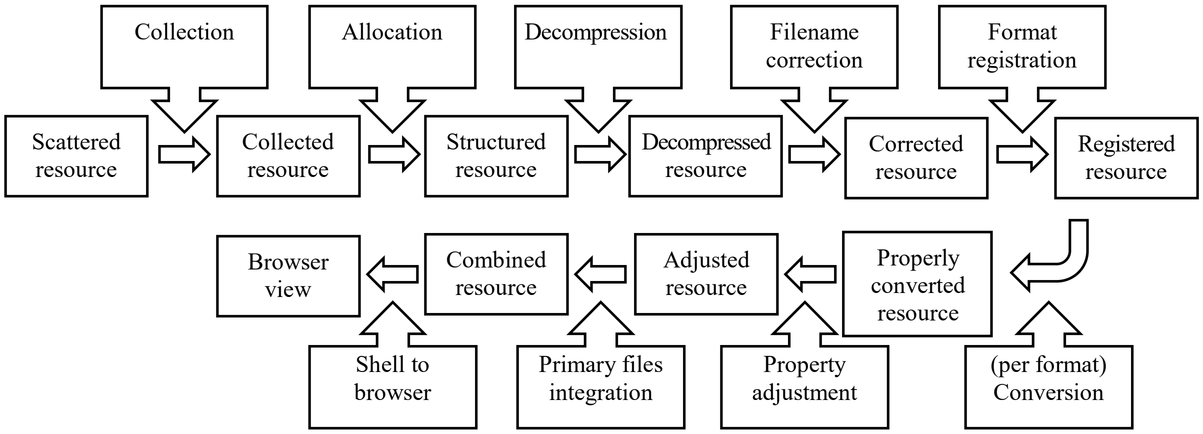


Figure 29: Overall process and related data.