XML Data Model Visualization

Syltinsy P. Jenkins
Department of Computer Science
Bowie State University
Bowie, USA
jenkinss0218@students.bowiestate.edu

Palmer Young
Department of Computer Science
Bowie State University
Bowie, USA
pcy12@yahoo.com

Abdullah Algahtania
Department of Computer Science
Bowie State University
Bowie, USA
algahtania0407@students.bowiestate.edu

Dr. Sharad Sharma
Department of Computer Science
Bowie State University
Bowie, USA
ssharma@bowiestate.edu

***Abstract*—The VR XML Data Model Visualization Project is a python based virtual reality project that generates 3D tree-graph representation of XML data in an immersive and interactive virtual environment.**

***Keywords*— virtual reality, VR, data model visualization, XML model**

I. Introduction

Data models are often represented as spreadsheet tables or eXtensible Markup Language (XML) files. The relationships are sometimes visualized using Unified Modeling Language (UML) and related software architecture tools. However, they are often flattened and limited views. An interactive 3D representation provides a new and different way of looking at and manipulating models which can foster interest in data architecture and data modeling.

This project was created under the direction of Dr. Sharad Sharma for the Bowie State University Computer Science Course COSC 729: Virtual Reality and its Applications.

II. Goal and Objectives

The purpose of this project is to create an interactive 3D representation of a data model in a virtual environment to foster interest in data architecture and data modeling. The target audience is for anyone to begin exploration of a new modality for manipulating data. This project also aims to demonstrate some fundamentals of virtual reality (VR) through creation the virtual environment and was designed with intent that this project would be extended and continued.

Due to time limitations, this project is currently still in its developmental phase and thus is used for demonstration purposes and can be used by other experienced software developers to: create 3D interactive visualizations for XML-based data models, view code examples for using Vizard platform features and other code libraries. This project demonstrates concepts covered in the course including avatar manipulation, use of sensors, sound control, and various user interface features including keypresses, mouse controls, and onscreen menus.

Concepts outside the direct course instruction include additional manipulation of objects, ability to read in and write out external files, and the ability to relate external data to virtual objects.

III. Project Description

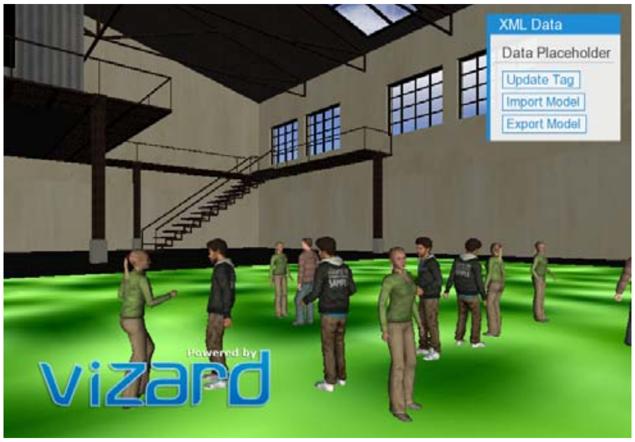
The VR XML Data Model Visualization Project was created using the Vizard platform along with several graphical modeling tools including SketchUp and 3DS Max.

The centerpiece of the virtual environment is the data model itself which is a 3D tree-graph representation of the XML data. The model is generated in a three-step process. First the data model is read in form an XML file. Then recursive methods are used to walk the data to generate the 3D tree graph. Trigonometry was used to create placement for the nodes in 3D. Node size and colors are used to express different data types and attributes contained in an XML data file.

IV. Functionality

The scene is set in a large warehouse-like facility. In the opening scene as shown in Fig. 1, a group of people are leisurely talking and dancing in the center area while dance music is playing in the background.

Fig. 1. Introductory scene: group of people in main area.



The user presses the ‘c’ key to clear the area. Pressing ‘c’ will pause the dance music playing in the background, play an audio file announcing that participants should ‘Clear the area for a demonstration.’ The characters in the scene will then walk to an observation area to watch when a model is shown.

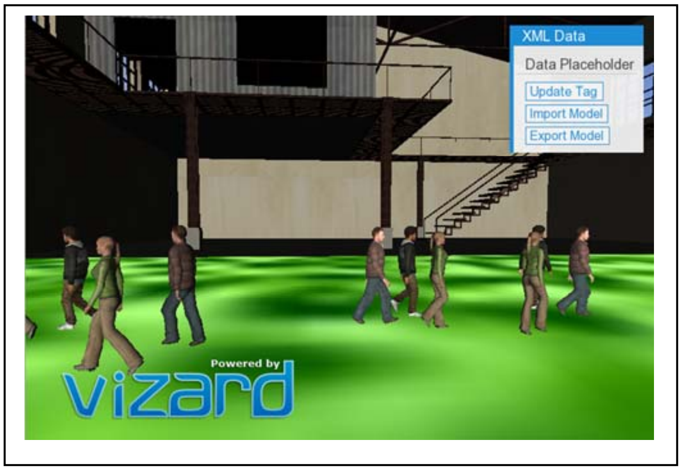


Fig. 2. Chracters clear the area.

A. *Primary User Controls*

After the characters move to an observation area, the user can click the 'Import Model' button in the 'XML Data' information menu. This will load the 'country_data.xml' file. As shown in Figure 3 below, in the 'countries.xml' file, node size was used to reflect number of contained nodes such that larger nodes contain more child nodes and the smallest nodes are leaf nodes containing no child nodes. Red nodes were used to represent the primary data type 'country' while blue was used to represent the 'year 'that the country was founded. Green was used to indicate economic data for gross domestic product (GDP). Finally, links are added to show the relationship between parent and child data nodes.

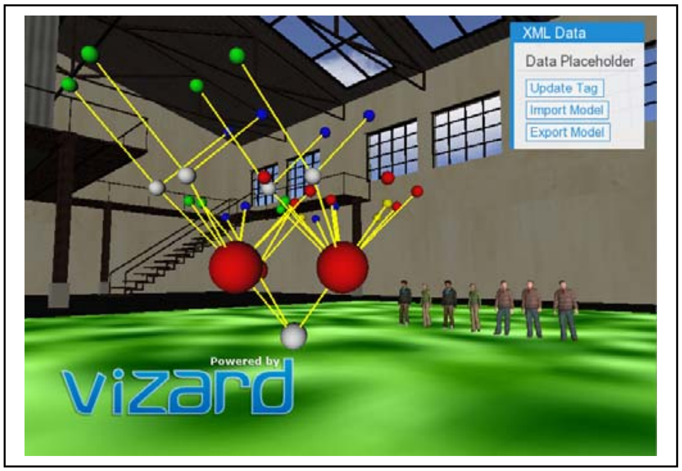


Fig. 3. Import the model.

Once the user imports the model, they will have the capabiilty to use the Grabber to grab any Node from the Data Model and move it anywhere with the VR Environment. As shown in Figures 4 and 5. Figure 4 shows 'Grabbing' a Node. Vizconnect configuration and in-game sensors are used to detect when the Grabber makes contact with the Node. User controls for 'double-click' to 'Create New Node,' 'spacebar,' to delete node, and 'Left-click and drag' to move nodes.

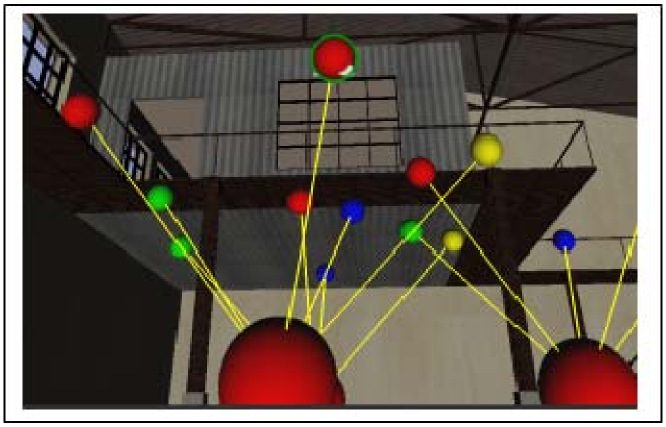


Fig: 4: Grabbing a Node.

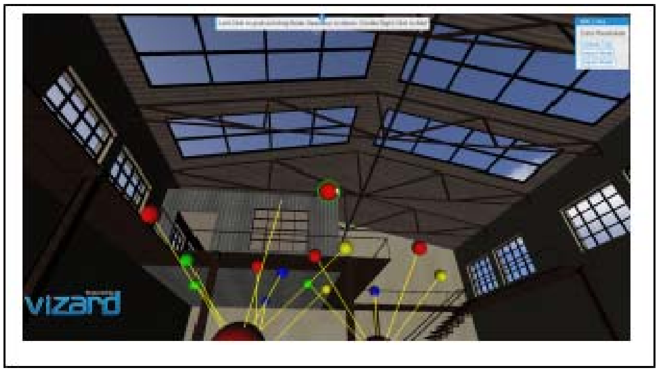


Fig. 5. Moving a Node.

Furthermore, a list of controls will apper in the Center Control Panel providing the user with a list options on how to manipulate the node. The user will have the capability to delete and re-add and attached the deleted node. As seen in Figures 6 and 7 below.

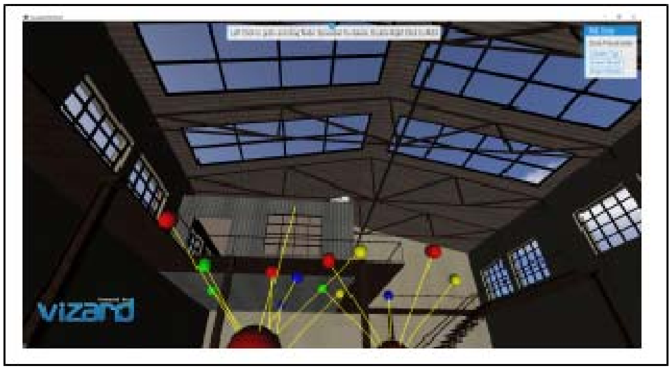


Fig. 6. Delete a Node.

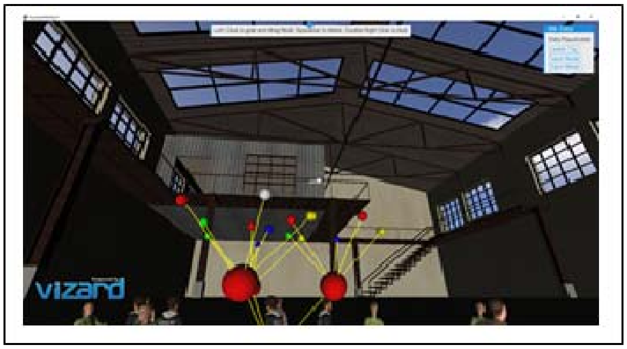
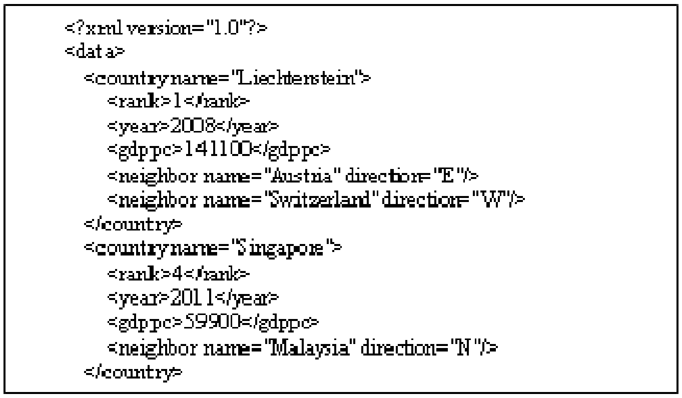


Fig. 7. Add Node.

Adding a Node can work in more than one way. The first is to replace a previously deleted node. The second is to create an arbitrary node. Although not fully implemented in this project, creation of a new node was intended to be able spawn creation of a separate model and XML file.

B. *Explorining Multiple Data Models*

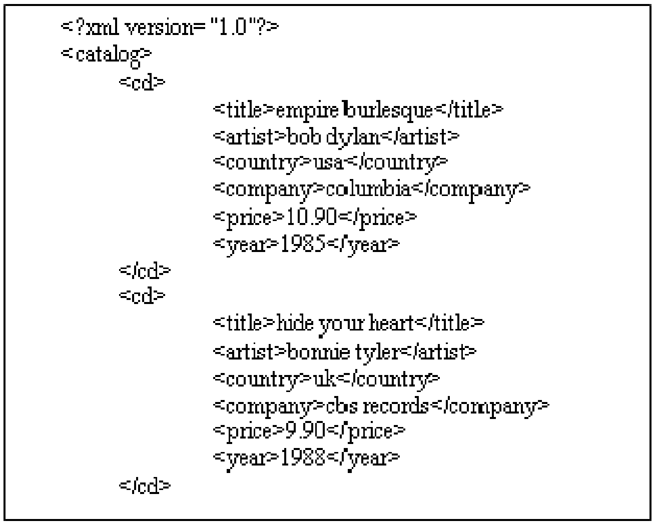
The XML visualization code allows the user to be able to render more than one model in the environment at a time. To illustrate this feature, two additional data models can be added to the environment by pressing the ‘8’ key to load the ‘cd_catelog.xml’ file and pressing the ‘9’ key to add the ‘FACE_SDM.face’ file. An excerpt of the source of the central model, from ‘country_data.xml’ is as follows in Figure 8 below.



```
<?xml version="1.0"?>
<data>
  <countryname="Liechtenstein">
     <rank>1</rank>
     <year>2008</year>
     <gdppc>141100</gdppc>
     <neighbor name="Austria" direction="E"/>
     <neighbor name="Switzerland" direction="W"/>
  </country>
  <countryname="Singapore">
     <rank>4</rank>
     <year>2011</year>
     <gdppc>59900</gdppc>
     <neighbor name="Malaysia" direction="N"/>
  </country>
```

Fig. 8. Code exerpt from country_data.xml.

The ‘cd_catalog.xml’ data source in comparison is shown in Figure 9. The ‘country data’ and ‘cd_catalog.xml’ files are similar in structure in terms of using tradtional concepts as XML tag names.



```
<?xml version="1.0"?>
<catalog>
      <cd>
            <title>empire burlesque</title>
            <artist>bob dylan</artist>
            <country>usa</country>
            <company>columbia</company>
            <price>10.90</price>
            <year>1985</year>
      </cd>
      <cd>
            <title>hide your heart</title>
            <artist>bonnie tyler</artist>
            <country>uk</country>
            <company>cbs records</company>
            <price>9.90</price>
            <year>1988</year>
      </cd>
```

Fig. 9. Code exerpt from cd_catalog.xml.

The structual as well as content similarities and differences in the model are more easily seen in the 3D virtual representation as shown in Figure 10. The 3D model on the left is the ‘CD Catalog Data Model’ while the model on the right is the ‘Country Data Model.’

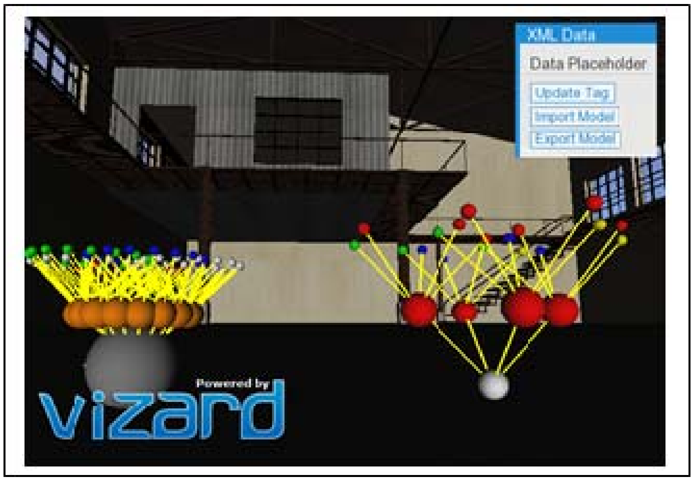


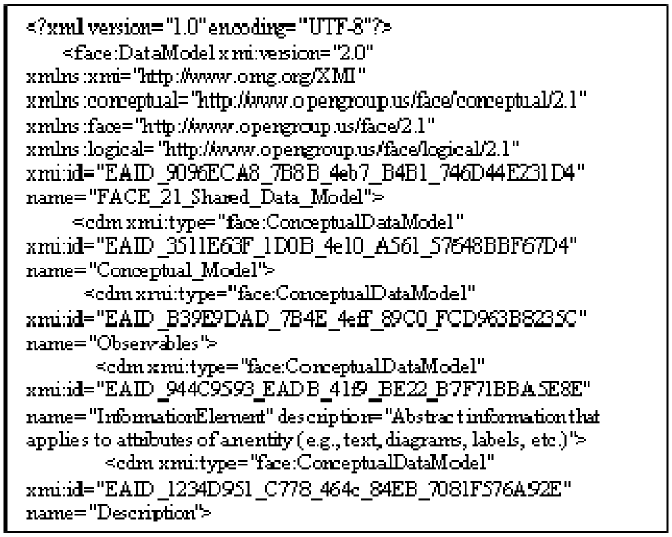
Fig. 10. cd_catelog.xml and country_data.xml 3D models.

Figure 10 shows the uniform structure of the CD Catalog Data Model entries as the shape of the model is regular indicating a consistent number of data fields only one level deep as reflected from the code excerpt in Figure 9. The Country Data Model is not symmetric meaning that the data has different numbers of nested child nodes. In the Country Data Model, there are varying numbers of neighbor nodes per country and in some cases, additional inner nodes.

Also note that there are similar node colors. Red indicates country name information, green represents monetary information. In the case of the country data, monetary information is GDP, while in cd catalog data, monetary information is price of a cd.

Blue represents date and time information: date founded for country data, and date published for cd catalog data . This enables user's to quickly and easily see where data could be cross-referenced.

The third model, the 'FACE_SDM.face' file is a different type of XML data model. It is a semantic relational model used to map different concepts together. Figure 7 shows a sample source code sample from this model.



```
<?xml version="1.0"encoding="UTF-8"?>
     <face:DataModel xmi:version="2.0"
xmlns:xmi="http://www.omg.org/XMI"
xmlns:conceptual="http://www.opengroup.us/face/conceptual/2.1"
xmlns:face="http://www.opengroup.us/face/2.1"
xmlns:logical="http://www.opengroup.us/face/logical/2.1"
xmi:id="EAID_9096ECA8_7B8B_4eb7_B4B1_746D44E231D4"
name="FACE_21_Shared_Data_Model">
      <cdm xmi:type="face:ConceptualDataModel"
xmi:id="EAID_3511E63F_1D0B_4e10_A561_57648BBF67D4"
name="Conceptual_Model">
        <cdm xmi:type="face:ConceptualDataModel"
xmi:id="EAID_B39E9DAD_7B4E_4eff_89C0_FCD963B8235C"
name="Observables">
          <cdm xmi:type="face:ConceptualDataModel"
xmi:id="EAID_944C9593_EADB_41f9_BE22_B7F71BBA5E8E"
name="InformationElement" description="Abstract information that
applies to attributes of an entity (e.g., text, diagrams, labels, etc.)">
           <cdm xmi:type="face:ConceptualDataModel"
xmi:id="EAID_1234D951_C778_464c_84EB_7081F576A92E"
name="Description">
```

Fig. 11. FACE_SDM.face code exerpt.

When comparing this model to ther other models, we note that the size shape, colors, and structures are very different from the simple tradional XML tag information. The this semantic and relationship model is irregular with varying numbers of attributes, fields, and child nodes per node.

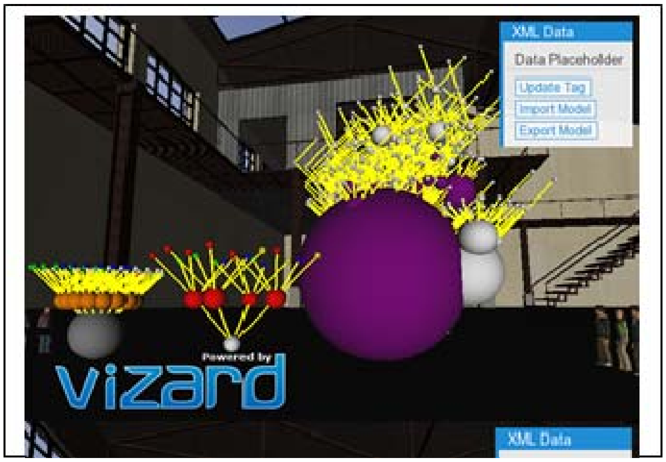


Fig. 12. Cd_catelog.xml, country_data.xml, and FACE_SDM.xml 3D VR data models.

To glean more information from this very large and complex data model, additional methods such as data conditioning, filtering, and sorting would be required to better understand and examine the structure of the model. [1]

V. Limitations and Lessons Learned

The Vizard platform is an excellent rapid development integrated development environment (IDE) for VR. However, to manipulate outside data, other software libraries must be used. Standard python can be used. In this application, the Element Tree XML Application Programming Interface (API) was used for parsing the XML data files.

There are several areas for future growth for this project. Due to time constraints, there were several aspects of this project that require further development for completeness. These include real-time updating for moving connected links in real-time, updating and changing links. The current state of this project requires experienced software developers to create a guided experience for users.

The development plan underestimated the level of effort required to complete this project. It took many hours of practice to learn to manipulate models and understand the features of the VR platform, as well as the Element Tree API to produce this project. Model placement and manipulation of coordinates were difficult to view in the Vizard IDE. Recursion and mathematics are required to create a reasonable arrangement of nodes to produce a truly 3D model. Configuration files also must be managed and the interface for Vizconnect to enable features such as grabbing and avatar control when performing collaborative development.

VI. Future Application

3D and VR data visualization is a new and growing area of research. Advancements for this project would include multiuser support, hardware support, hardware support, and storyboard application.

Multiuser support, which requires network capability, would enable multiple users to view and manipulate the model. Hardware support including head-mounted display such as Oculus and input devices such as pinch glove, will provide for a more immersive experience.

Storyboarding is about how the application itself can be used. For example, data scientists can use this project to explore and different views of information to others. This could also be used for data model training.

Next steps would creation of an interactive, guided tutorial on how to build data models in the 3D environment. Adding VR hardware support will make the model more interactive and can be leveraged to provide visual and haptic feedback when incorrect connections are made.

Other applications include opportunities for human factors studies to explore the effects of node versus link size, shape, and color and measures of attentiveness and duration of game play.

VII. Conclusion

This project provided a basic demonstration of a 3D VR representation of XML data models 3-dimensional tree-graphs. Multiple XML documents were imported and rendered as models in the virtual environment. This project exercised some fundamentals of creating an immersive VR environment through use of on-screen prompts, use of different types of sounds, VR sensors, timers, animations and avatars on the Vizard platform. Recursion and trigonometric functions were used to create patterned structures while color, links, and node sizes were used to illustrate differences in datatypes and relationships between nodes.

VR is a good medium for gaining a new perspective on visualizing structure of data models. VR IDEs such as Vizard can be leveraged for rapid VR prototyping with flexibility to bring in external code libraries. The immersive environment makes data visualization more interesting not jut to the user, but also to the developers. Team members look forward to continuing this work.

VIII. References

[1] Ms90Prod, Composer, *Reggaeton vs Rap.* [Sound Recording]. looperman. 2019.

[2] B. Shneiderman, C. Plaisant, M. S. Cohen, S. M. Jacobs and N. Elmqvist, Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th ed., Hoboken, NJ: Pearson Education, Inc., 2017.

[3] WorldViz, "Vizard 6 Documentation," WorldViz, 2019. [Online]. Available: https://docs.worldviz.com/vizard/latest/. [Accessed 17 April 2019].

[4] Python Software Foundation, "The ElementTree XML API," Python Software Foundation, 18 March 2019. [Online]. Available: https://docs.python.org/2/library/xml.etree.elementtree.html. [Accessed 24 April 2019].

[5] S. P. Sharad, "COSC 729: Virtual Reality and its Applications," Bowie State University, April 2019. [Online]. Available: http://www.cs.bowiestate.edu/sharad/. [Accessed 17 April 2019].