A Conceptual View Of The Building Blocks To Develop A Web-Based Training And Educational Tool

Tulio Sulbaran *Ph.D*[1]

Abstract – For years, primitive tools such as text and 2D drawings, sketches, graphs and figures have been used for training and education in many areas. With the improvement of technology, a new tool known as virtual reality was born with the idea that perhaps computers could be used to digitally display information three dimensionally. Since its inception, virtual reality has been used successfully for many applications such as flight simulations, medical training, scientific simulations, education and more.

Recently, virtual reality has made a significant impact on the Internet. People are able to not only create virtual worlds or environments, but place them over the World Wide Web so that others may view them and even interact with them within their Internet browser. The Virtual Reality environments can also be used in conjunction with other technologies, namely database and programming languages, to enhance functionality. A user of the combination of these technologies can be absorbed into a VR environment, interact with it, and the user's input can be recorded into a database.

This paper provides information about the most common features of virtual reality modeling language, current database management tools, and current programming languages available to the public. This paper also provides a conceptual view on how virtual reality modeling language, Microsoft SQL Server 2005 database management software, and Java programming can be used collectively to create a Web based virtual reality training and educational tool. It is anticipated that this paper will help the advancement of scholarship in engineering education by providing a foundation for a variety of teaching tools using this or similar technologies.

Keywords: Web-Based, Virtual Reality, Education, Tool

INTRODUCTION OF VIRTUAL REALITY

Virtual reality or VR, as it is better known to some, has been difficult to define through the years. The term virtual reality was coined by Jason Lanier and he describes virtual reality as an immersive, interactive simulation of realistic or imaginary environments (Brady & O'Sullivan, 1999). John C. Briggs (2002) believes that "VR is best defined as a computer-generated 3-D experience in which a user can navigate around, interact with, and be immersed in another environment or world in real time, or 'at the speed of life'." There are others who believe that one can experience virtual reality by simply watching a movie, reading a book, or just using the imagination and many believe that virtual reality is something that can only be experienced through the use of expensive technical equipment such as Head Mounted Displays (HMD), Glove Input Devices, and large immersive systems such as a multi wall immersive ICUBE display. Although these are fine definitions for virtual reality, R.S. Kalawsky (1996) states that "the definition of a VR system lies with a description of what the system enables one to do rather than basing the definition on a technology based description". Kalawsky (1996) also believes that the term 'VR' should be taken in a broader context and by doing so it allows a participant to:

- Become immersed in a completely synthetic computer generated environment.
- Achieve a sense of presence in the environment.

[1] Associate Professor – School of Construction at the University of Southern Mississippi, Box 5138, Hattiesburg, MS, 39406. E-mail: Tulio.Sulbaran@usm.edu.

- Become un-inhibited where conventional laws of physics can be controlled in a way that assists greater understanding.
- Achieve a sense of non-real time; where situations can be presented in slow or fast time.
- Achieve a high degree of interaction that can equal or exceed that achievable in the real world.
- Interact in a completely natural and intuitive manner with the synthetic environment.
- Repeat the task until the desired level of proficiency or skill has been achieved.
- Perform in a safe environment

To simplify the virtual reality system, Kalawsky (1996) also designed a model for what he called a "Generic Model of a Virtual Environment System" (Shown in Figure 1).

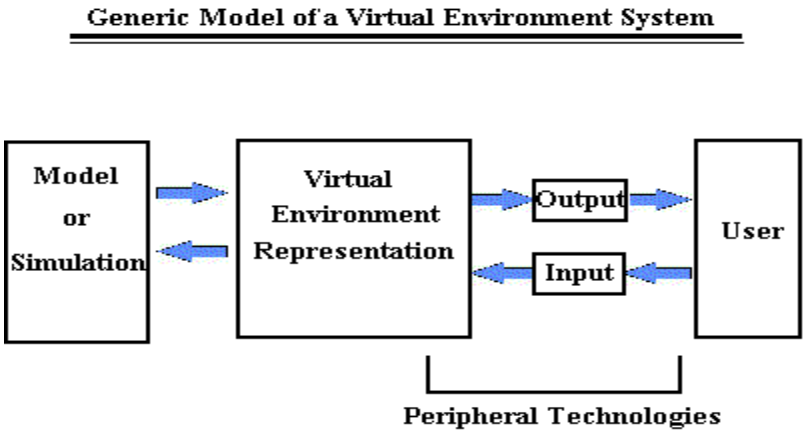

Figure 1. Kawalsky's Generic Model of a Virtual Environment System

Brady and O'Sullivan (1999) explain Kawalsky's model by stating that "when a person interacts with a Virtual Environment (VE), they become engaged in a closed loop. The user interacts with the VE representation, which in turn is based on a model. The effects of this interaction are fed back to the user through the modification of the model based on the user's input and so the cycle begins again".

VRML, also known as Virtual Reality Modeling Language or vermal, is a standard in the development for 3D on the Web. According to the Web3D Consortium website (2005), "The Virtual Reality Modeling Language (VRML) is a file format for describing interactive 3D objects and worlds. VRML is designed to be used on the Internet, intranets, and local client systems. VRML is also intended to be a universal interchange format for integrated 3D graphics and multimedia. VRML may be used in a variety of application areas such as engineering and scientific visualization, multimedia presentations, entertainment and educational titles, web pages, and shared virtual worlds." Although it is possible to produce a Web based VR interactive world, user interactivity is very limited using VRML alone. Some of the interactive features of VRML alone include navigation, viewpoints, special nodes, and sensors.

There are many options when it comes to navigating through a VRML environment. These options include walk, fly, and examine or study. Turn and pan are also options in navigation but are sub-options of walk, fly, and examine or study. When walking through a VRML world, the user is able to walk around at a pre-defined ground level inside the environment. The user is able to turn the first person view to the left, right, up, or down and they are able to pan which means to move left or right without changing the current view. When using fly mode, the user is able to fly around the virtual reality environment in every direction which means the user can walk to an object and instead of walking around it, the user can go into fly mode and take to the virtual air above the object and land on the other side. The turn and pan options are also available in fly mode. When the user is in the examine or study mode, he or she is able to view every object in the VRML world as if they were holding it in their hands and rotating the entire world in different directions.

Viewpoints in a VRML environment are camera positions that are pre-defined by the creator of the VRML world. The creator can place as many viewpoints in a world that is necessary and give each a description. Viewpoints can be used in many different ways. One way to use viewpoints is by placing them around a certain object so that the user can see the object from different angles. Another way is to place viewpoints in different rooms of a building created by VRML so that the user is able to jump very quickly from one room to another without having to navigate through the entire building to find a specific room. Viewpoints are also very useful in large VRML environments because they act as transporters for the user. The user is able to travel a long way to get to a certain area of the world without having to walk the entire distance.

There are many special nodes available in VRML that can add interactivity and realism to the world one has created. Some of these nodes include switch, billboard, anchor, background, and fog. A switch node provides the use of triggers that allows the user to switch from one item to another when they click on an object. A billboard node allows an object to automatically rotate itself so that it is always aligned with the user's viewpoint. An anchor node, similar to an HTML hyperlink, embeds a hyperlink specified by the author into an object so that when a user clicks the object, the specified URL opens in the browser. A background node provides a background in two ways: gradient colors or textures. When using the gradient colors option to add a background, the creator chooses a color for the sky and the ground. The other choice is using realistic textures and applying them to an infinitely large six-sided cube also known as a Sky Box. The six sides are up, down, north, east, west, and south. For example, you could apply a realistic texture of the sky for the up direction of the Sky Box, a texture of ground for the down direction, and textures of the horizon from the north, east, west, and south directions. When the user views the world, he or she will experience a seamless horizon that surrounds them. The fog node is used simply to add a realistic fog or mist to the VRML environment. The attributes of the fog node are the color, fog type, and visibility range. The color specifies the fog color and the visibility range determines how far the user can see an object in the fog. Once the object is past the range, it becomes the color of the fog. The fog type can have two values: linear and exponential. Linear fog affects depth perception by becoming more linear with distance while exponential makes the fog look more natural.

VRML sensors that can be utilized by the creator of the VRML world include touch sensors, visibility sensors, dragging sensors, and proximity sensors. All of these sensors provide more user interactivity in the VRML environment. The touch sensor is usually defined in a group and affects all object placed within that group. The sensor reacts when the user moves the mouse over an object in the group and when the user clicks an object in the group. When clicked, the sensor can generate an event such as playing a sound. Visibility sensors can generate events such as playing a sound or even enabling another sensor when a particular object becomes visible to the user in the VRML environment. Dragging sensors are a special sensor that allows a user to click on an object and move it inside the world. There are three drag sensor options: plane, cylinder, and sphere. The plane sensor allows an object to be dragged along the XY plane. The cylinder sensor plots the object's movement to that of a conceptual cylinder and the sphere sensor plots the object's movement to that of a conceptual sphere. Proximity sensors allow the author of the VRML environment to define an invisible rectangular box. When the user enters, leaves, or moves in the hidden box, an event can be generated such as playing a sound or even switching a light node from on to off.

Although these basic features allow one to create an amazing VRML world, there are many things VRML is incapable of doing alone.

When building a training and educational tool in VRML, it might be required that a database be used to store information relating to a user of the tool or perhaps the user's performance concerning the tool. Another requirement might be that the training tool be a server-side Web application. Requirements of this nature can not be accomplished through VRML alone.

If the author of the training tool requires information to be stored relating to the user of the tool and user performance, a database is required. VRML alone is incapable of communicating with a database so the author must use programming to establish the connection. Java is a simple, object-oriented, dynamic language that can be utilized to establish the database connection. Java servlets can also be used to fulfill the requirement of making the training tool a server-side Web application. When database, Java, and VRML are brought together, a very powerful and useful Web based VR educational and training tool can be developed.

DATABASES

A simple definition of the term database is an organized body of related information that is stored on a computer. Another more specific definition of a database would be a collection of data that is organized so that its contents can easily be accessed, managed and updated. There are many definitions used to describe a database as well as many types of databases and database management software. The most popular type of database is called the relational database. A relational database conforms to a relational model and is based on the arrangement of the data in the database. The relational database model allows files to be related by means of a common field. In order to relate any two files, a common field is required. This makes the relational database model extremely flexible. An example of a relational database would be creating a database called Library with two tables: Author and Book. Inside the Author table, three columns are defined: ID, First Name, and Last Name. Inside the Book table, three columns are also defined: Author ID, Title, and Publish Year. The common field between these two tables that relates them is ID and Author ID. Please refer to the figure below for a graphical representation of the two related tables.

Author

ID	First Name	Last Name
1	Stephen	King
2	Tom	Clancy
3	Michael	Crichton

Book

Author ID	Title	Publish Year
1	Christine	1983
2	Debt of Honor	1995
3	Jurassic Park	1991

Figure 2. Graphical Representation of the Relational Database: Library.

There are many relational database management tools available for use. Some of the most popular or well-known relational database management tools are MySQL, Microsoft SQL Server 2005, Oracle, DB2, and Informix. Each has their own unique characteristics such as which operating systems they support and the type of software license available. The type of software license supported by most mass produced relational database management tools is proprietary. A proprietary license means that there are restrictions on using, copying and modifying as enforced by the proprietor. In simpler terms, a proprietary license means the product is not free. Some software, such as MySQL, supports both proprietary and General Public Use (GPL) licenses. A GPL license is sometimes referred to as a copyleft license for software. This means that people are free to make modifications or upgrades to the software as long as they also support a GPL license when something new is created. This prevents someone from creating new software from old software and then copyrighting and selling the newly created software. Please refer to the table on the next page for some general and technical information about the five relational database management tools: MySQL, Microsoft SQL Server 2005, Oracle, DB2, and Informix.

Table 1. General and Technical Information Comparison of Some Relational Database Management Systems

Database Software	**Maintainer**	**Software License**	**Operating System Support**
MySQL	MySQL AB	GPL or Proprietary	Windows, Mac OS, Linux, UNIX, maybe z/OS (IBM)
Microsoft SQL Server 2005	Microsoft	Proprietary	Windows
Oracle	Oracle Corporation	Proprietary	Windows, Mac OS, Linux, UNIX, z/OS (IBM)
DB2	IBM	Proprietary	Windows, Linux, UNIX, z/OS (IBM)
Informix	IBM	Proprietary	Windows, Mac OS, Linux, UNIX

The researcher found that the best software to use when creating databases for a virtual reality training and educational tool was Microsoft SQL Server 2005. "The SQL Server 2005 database engine provides more secure, reliable storage for both relational and structured data, enabling you to build and manage highly available, performant data applications…." (Microsoft, 2005). Microsoft (2005) also lists the following strengths concerning the database management software:

- Build, deploy, and manage enterprise applications that are more secure, scalable, and reliable.
- Maximize IT productivity by reducing the complexity of developing and supporting database applications.
- Share data across multiple platforms, applications, and devices to make it easier to connect internal and external systems.
- Control costs without sacrificing performance, availability, scalability, or security.

Microsoft SQL Server 2005 also provides the user with a comprehensive feature set, interoperability with existing systems, and automation of routine tasks. The software includes many management tools that allow the user to create and maintain a database system easily (Shown in Figure 3).

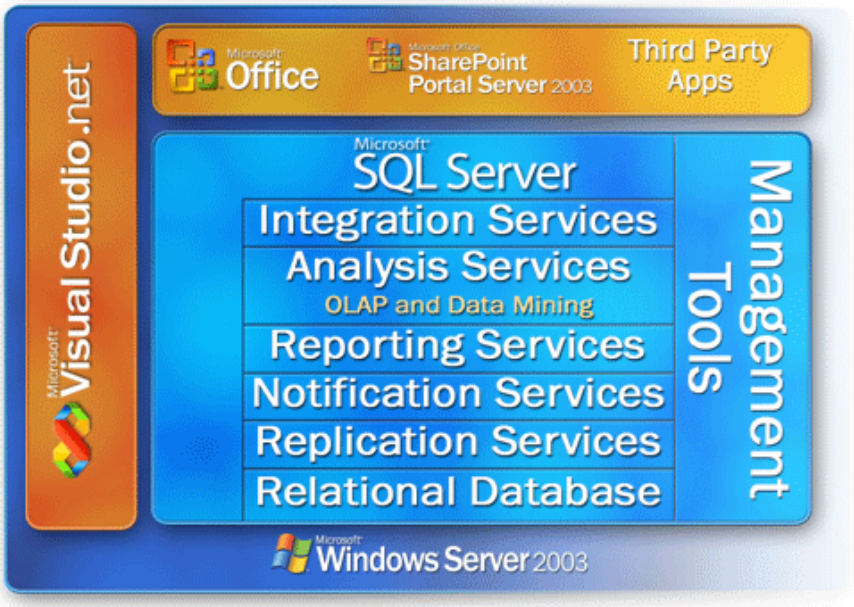

Figure 3. Layout of the SQL Server 2005 Data Platform

The SQL Server data platform includes the following tools (Microsoft, 2005):

- **Relational Database**. A more secure, reliable, scalable, highly available relational database engine with improved performance and support for structured and unstructured (XML) data.
- **Replication Services**. Data replication for distributed or mobile data processing applications, high systems availability, scalable concurrency with secondary data stores for enterprise reporting solutions, and integration with heterogeneous systems, including existing Oracle databases.
- **Notification Services**. Advanced notification capabilities for the development and deployment of scalable applications that can deliver personalized, timely information updates to a variety of connected and mobile devices.
- **Integration Services**. Data extraction, transformation, and loading (ETL) capabilities for data warehousing and enterprise-wide data integration.
- **Analysis Services**. Online analytical processing (OLAP) capabilities for the rapid and sophisticated analysis of large and complex datasets using multidimensional storage.
- **Reporting Services**. A comprehensive solution for creating, managing, and delivering both traditional, paper-oriented reports and interactive, Web-based reports.
- **Management Tools**. SQL Server includes integrated management tools for advanced database management and tuning as well as tight integration with tools such as Microsoft Operations Manager (MOM) and Microsoft Systems Management Server (SMS). Standard data access protocols drastically reduce the time it takes to integrate data in SQL Server with existing systems. In addition, native Web service support is built into SQL Server to ensure interoperability with other applications and platforms.
- **Development Tools**. SQL Server offers integrated development tools for the database engine, data extraction, transformation, and loading, data mining, OLAP, and reporting that are tightly integrated with Microsoft Visual Studio to provide end-to-end application development capabilities. Every major subsystem in SQL Server ships with its own object model and set of application program interfaces (APIs) to extend the data system in any direction that is unique to your business.

The researcher also finds the software's graphical user interface (GUI) to be very organized and easy to follow. The software's SQL Express Manager makes it very simple for the user to edit queries and an explorer window to the left side of the GUI allows the user to easily navigate through the server's directories and connect to the proper database files (Shown in Figure 4).

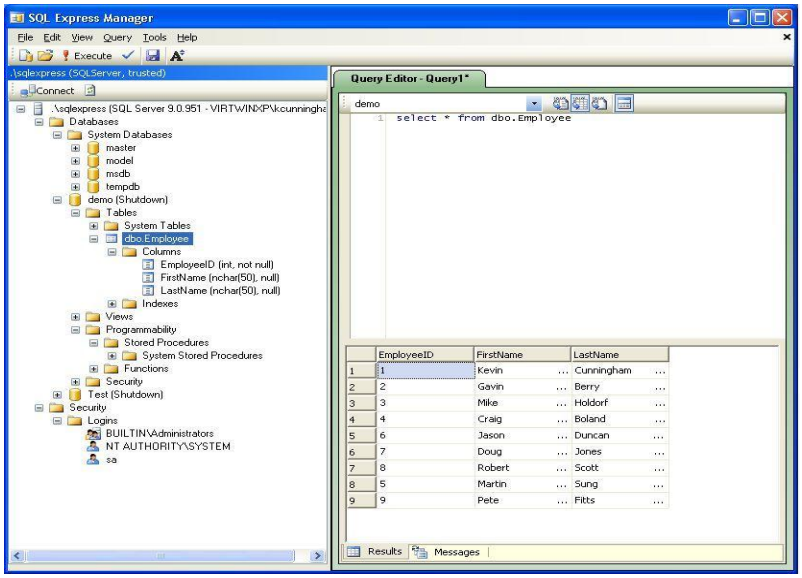

Figure 4. A screenshot of Microsoft SQL Server 2005's SQL Express Manager GUI

By using Microsoft SQL Server 2005's secure and reliable relational database engine and easy to follow GUI and management tools, the necessary databases needed to accommodate a Web based virtual reality training and educational tool can be easily built.

PROGRAMMING

A programming language can be described as a language composed of a specific vocabulary and specific grammatical standards that can instruct a computer to perform a specific task. Another yet shorter definition for programming language is high-level language. There are many different high-level languages available for use and new ones are being created constantly. Some of the most popular programming languages include Java, C, C++, Perl, and Visual Basic. Each language has its own unique features such as keywords and syntax for organizing instructions. Please refer to the table below for a features comparison between Java, C, C++, Perl, and Visual Basic programming languages.

Table 2. Programming Language Features Comparison of Java, C, C++, Perl, and Visual Basic

Features	*Java*	*C*	*C++*	*Perl*	*Visual Basic*
Object-Orientation	Hybrid	Hybrid	Hybrid/Multi-Paradigm	Add-On/ Hybrid	Partial Support
Static/Dynamic Typing	Static	Static	Static	Dynamic	Static
General Classes	No	No	Yes	N/A	No
Inheritance	Single class, multiple interfaces	Single class, multiple interfaces	Multiple	Multiple	None
Method Overloading	Yes	Yes	Yes	No	No
Operator Overloading	No	Yes	Yes	Yes	Yes
Garbage Collection	Mark and Sweep or Generational	Mark and Sweep or Generational	None	Reference Counting	Reference Counting
Class Variables/ Methods	Yes	Yes	Yes	No	No
Access Control	public, protected, "package", private	public, protected, private, internal, protected internal	public, protected, private, "friends"	None	public, private
Regular Expressions	Standard Library	Standard Library	No	Built-in	No
Language	C, some C++	All .NET	C, Assembler	C, C++	C (via DCOM)

Integration		Languages			
Built-In Security	Yes	Yes	No	Yes (perlsec)	No

When building a Web based training and educational tool, the researcher discovered that using Java and Java servlets was a wise decision. The Java Web site defines Java as "a simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language" (The Java Language: An Overview, 2005). What makes the java language simple is that it can be "programmed easily without a lot of esoteric training" (The Java Language: An Overview, 2005). Another way the Java language is simple is it's automatic garbage collection (the freeing of memory not being used) which makes programming tasks easier and cuts down on the appearance of bugs. The Java language is also object-oriented and the Java Web site states that this is "powerful because it facilitates the clean definition of interfaces and makes it possible to provide reusable "software ICs [integrated circuits]" (The Java Language: An Overview, 2005). In simpler terms, Java focuses more on the application being created and less on the code that built it. Java is also network-savvy meaning it can deal easily with TCP/IP protocols such as HTTP and FTP. The Java Web site (2005) goes into more detail by stating that "Java applications can open and access objects across the net via URLs with the same ease that programmers are used to when accessing a local file system." There are also Web based Java programs that can be written known as applets and servlets. The main difference between the two programs is that an applet runs inside a Web browser on the client's machine while a servlet runs on the Web server. Unlike an applet, a servlet can process user input and return a response to the user. "If a servlet is designed to provide information to the user, it can track down the information it needs in any number of ways -- by retrieving it from a database, for example. The servlet can then dynamically construct an HTML page containing the information it has collected, and can display that page in the user's browser" (Andrews, 2007). According to Paul S. Wang (2002), a Computer Science professor from Kent State University, the advantages of using the Java language can be summed up in the list below:

- Platform independence
- Totally Object-oriented, but simpler than C++
- Dynamic incremental loading and linking
- Automatic GC (garbage collection)
- Multithreaded
- Systematic class, package, and source file naming
- GUI and graphics programming
- Web and network applications support
- Internationalization (programs in Unicode)

For the reasons stated above, it is in a researcher's best interest to use java code for any scripting required to make a Web based training and educational tool work properly.

HTML or Hypertext Markup Language has been used in combination with Java to create powerful Web pages for use on the Internet. Like HTML, Virtual Reality Modeling Language or VRML can also be combined with Java to create a powerful Web based and interactive virtual environment for use on the Internet. VRML can also be combined with HTML. When combining VRML with HTML and Java, the objects can act out behaviors defined in the HTML and Java programming code that VRML alone can not accomplish. According to Bell and Carey (1997), there are six combinations in which someone can use VRML, HTML, and Java:

- VRML file inside an HTML file: This is a semi standard part of HTML using the <EMBED> or <OBJECT> HTML tag, although HTML does not require that HTML browsers support embedding of VRML files (or any other type of file) into HTML documents.
- Java code inside a VRML file: This is a standard (although not required) part of VRML 2.0, using a Script node that refers to the compiled Java code.
- Java applet communicating with a VRML browser: This is a not-yet-standard extension to VRML 2.0 known as the External Authoring Interface (EAI). At some time in the future it will probably become a standard (but perhaps not required) part of VRML.

- Java classes corresponding to VRML nodes: Several companies are developing programming toolkits that define in-memory representations of VRML nodes that can be used in any way the programmer wishes. These can be extremely useful when implementing VRML browsers or VRML tools, but none are yet a standard part of either VRML or Java.
- HTML file inside a VRML file: Using an HTML file as a texture map to display it inside a 3D world would be an interesting extension to VRML, but it is not yet supported by any VRML browser and is not part of VRML 2.0.
- Java applet inside a VRML file: Using a Java applet as a texture map to display the Java program inside the 3D world would also be an interesting extension. Interaction with the Java program could also be supported by projecting pointing device motion onto the applet. However, this also is not supported and is not part of VRML 2.0.

One example of combining VRML, HTML, and Java was in a project conducted by Szilard Kiss from the University of Twente in the Netherlands titled Web-Based VRML Modeling (2001). In this project, Kiss (2001) presented a method to connect VRML and Java components in a web page using EAI (external authoring interface) which makes possible to interactively generate and edit VRML meshes. With this program, a person could load a Java applet which contained buttons that allowed the altering of 3D meshes and then exporting the new model to VRML. The editing of the model was based on vertices and sets of vertices grouped in rings and columns (Kiss, 2001). The editing options that were created for the program dealt with the rings and columns of vertices including: adding rings, adding columns, enlarging rings or columns, reducing rings or columns, and smoothing a ring (Shown in Figure 5).

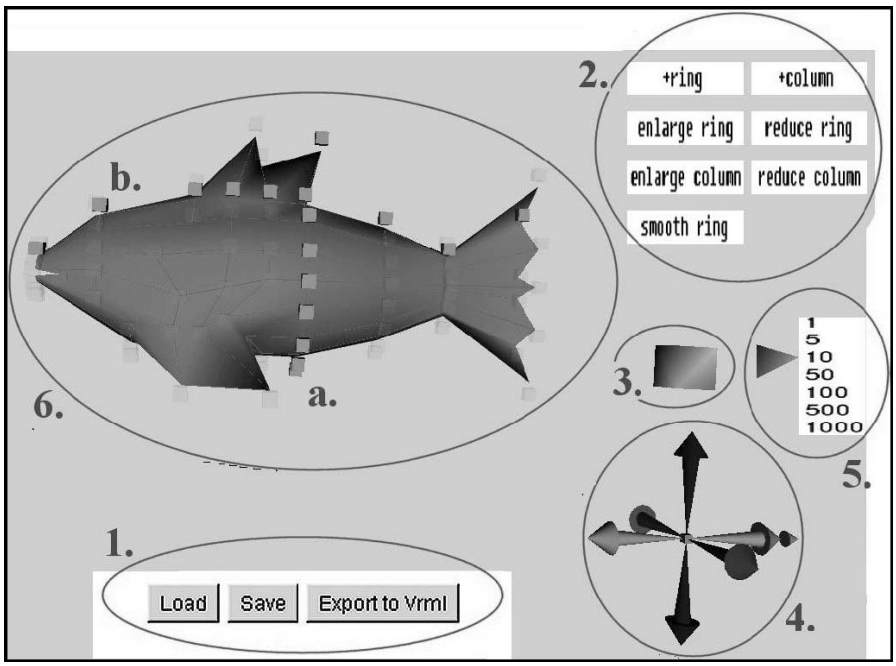

Figure 5. Screenshot of Kiss's VRML and Java external authoring interface (Kiss, 2001).

Any of the six combinations could be used when developing a Web based training and educational tool. The researcher recently built a Web based training tool that used the third combination described by Bell and Carey (1997): Java classes corresponding to VRML nodes. It worked well and the Java code, once executed, writes code to a VRML file and creates a unique Web based VR environment.

PUTTING THE BUILDING BLOCKS TOGETHER

In order to develop a Web based virtual reality training and educational tool, several tasks have to be performed. The following activities outline each step taken in planning and creating a VR training and educational tool prototype.

Step 1: Organizing the directory structure is very important to the functionality of a Web based VR training and educational tool. It is important in the beginning that the author creates a main directory on the Web server that will house all files pertaining to the VR training tool. This adds the advantage of centralizing all necessary files including VRML files, Java classes, and HTML files. It also adds the advantage of mobility so that if the Web based VR tool needs to be moved to another Web server, only slight modifications need to be made to the Java files that create the database connection. Please refer to the figure below for a sample flowchart of a directory structure.

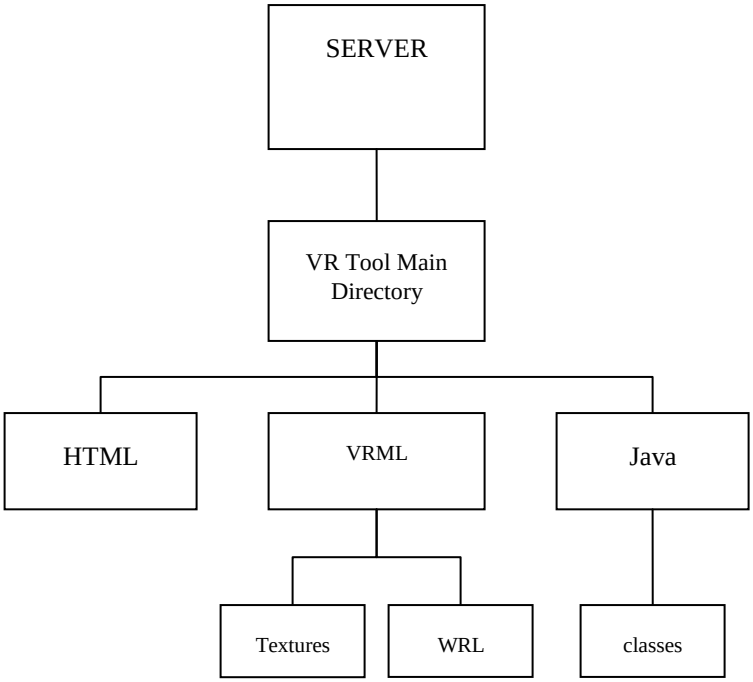

Figure 6. Sample directory structure of a Web based VR training tool.

Step 2: Using VRML coding or a VRML authoring toolkit, the researcher needs to create all necessary models that will be used for the training tool. During this process, the author needs to decide which models will be interactive and apply the necessary nodes such as switch, anchor, billboard, or sensors. The finished models should then be saved with the VRML file extension, .WRL.

Step 3: To ensure that the main VRML file to be generated for the training tool be relatively small in size, the author should use what is known in VRML as prototypes and external prototypes. Defining something as a prototype or PROTO is similar to creating your own VRML object that can be used over and over. External prototypes or EXTERNPROTO are prototypes defined in external files and referenced by a URL. The author should define all separate VRML files created as prototypes and then create a simple VRML output file that will reference them as external prototypes. The following table shows an example of defining an object as a PROTO and then referencing it as an EXTERNPROTO in another file:

Table 3. Defining PROTOs and EXTERNPROTOs

```
#VRML V2.0 utf8
#Model file
PROTO Model_1 [
] {
Group{
      children[
#*****************


#*****************
      ]#children
} #group


}#proto end
```

```
#VRML V2.0 utf8
#Output file
EXTERNPROTO Model_1[]"Model_1.wrl"
DEF a1 Model_1{}
```

This allows the output file, once generated by the user, to remain small in size even though several external prototypes can be called into it at once.

Step 4: The author of the training tool should then decide what databases or database is needed to store important information regarding the training tool and then decide what tables should be included in the database such as user information or user performance data. These databases can be created using Microsoft SQL Server 2005 software or any database management software deemed best by the author.

Step 5: The author needs to create any necessary Web pages for the VR training tool such as a login page, a welcome page, and Web pages that are referenced by anchor nodes if any are used. The author may do this using HTML coding alone or Web design software such as Microsoft FrontPage or Macromedia Dreamweaver. The login page would allow a user to register information including a username and password. The user could then login using that username and password and be brought to a welcome page. The welcome page would include a brief overview of the VR training tool, instructions on how to use it correctly, and any other information the author considers necessary. Perhaps a link to download and install a VRML plug-in necessary to view a VRML environment within an Internet browser could be included on the welcome page.

Step 6: The next step is creating Java classes that adhere to the Web based VR training and educational tool. The author, if not experienced in the language of Java programming, should seek assistance from someone who is experienced. The first Java file should include JDBC or Java Database Connectivity code so that a connection to the database will be established. Next Java code should be written that will store registration information provided by a user at a login page to the database. A Java class is also needed to be implemented when the user clicks a button to generate the VRML environment. When the button is pressed, a Java class is called that creates a VRML output file with extension .wrl and then writes text to this file that externally calls the VRML model files needed for the VR training and educational exercise. If anchor nodes are included in some of the VRML models and the Web pages require user input that should be stored in the database, a Java class is needed. This can be accomplished by creating a Submit button on the Web page and once the user inputs the information and clicks the button, a Java class is initialized to store the information to the database.

Step 7: The final step is testing the application. Many test runs of the project should be performed personally by the author. The test runs may include registering three anonymous users, logging in using one of the user's usernames, and then generating a VR world. After generating the VR world, the author should navigate through and check the appearance and functionality of all models within the VR environment. Once completed, the author should check the databases to ensure all of the information regarding user information and user performance is saved into the appropriate columns in the database tables.

Using these steps, an interactive Web based virtual reality training and educational tool can be created. One advantage to using these steps to put the building blocks of a Web based VR training and educational tool together is

that these steps be used to create several different VR tools. These steps can also be modified to create a simpler VR tool or to create an extended VR tool that encompasses more interactivity.

SUMMARY

Virtual reality has been defined in many ways, but the one characteristic that relates them all is immersion. In other words, a user of VR is somehow immersed or absorbed into a VR environment and is able to interact with it to enhance the experience. In order to build a 3D semi-immersive Web based virtual reality training and educational tool, VRML is necessary because it is the standard in the development for 3D on the Web. It also has many advantages including user interaction, easy navigation, and it is open source. This allows for the use of scripting to add behaviors to a VR world that cannot be done with VRML alone. When the user enters the world, other objects must appear in the environment that the user can interact with. In order for the tool to work properly, the researcher feels that the use of Microsoft SQL Server 2005 software to build the necessary databases and then using VRML in combination with Java and HTML to create interactivity is the correct path to follow. Using Java servlets with VRML and HTML will allow the VR training tool to connect to the database and record results of the VR training session so user performance can be evaluated later by an administrator.

ACKNOWLEDGEMENT

The author would like to recognize the valuable feedback given by the students.

REFERENCES

[1] Carey, R., & Bell, G. (1997). *The Annotated VRML 2.0 Reference Manual* [Electronic Version]. Retrieved October 23, 2010, from http://www.cs.vu.nl/~eliens/documents/vrml/reference/BOOK.HTM

[2] Microsoft. (2005, November 7). Servers. Application Platform. SQL Server. Product Information. Overview. Retrieved September 15, 2010, from http://www.microsoft.com/sql/prodinfo/overview/default.mspx

[3] Kiss, S. (2001). Web Based VRML Modeling. *Fifth International Conference on Information Visualisation (IV'01*), 612. Retrieved October 24, 2005, from IEEE Computer Society Digital Library database.

[4] Wang, P.S. (2010). Java Overview. *An Overview of Java*. Retrieved October 12, 2010, from http://sofpower.com/java/main.html

[5] The Java Language, An Overview. (2010). *Java*. Retrieved on November 15, 2010 from http://java.sun.com/docs/overviews/java/java-overview-1.html

[6] Wikipedia. (2010). Comparison of relational database management systems. Retrieved on March 1, 2010 from http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems

[7] Andrews, M. (2010). Story of a Servlet: An Instant Tutorial. Retrieved on January 25, 2010 from http://java.sun.com/products/servlet/articles/tutorial/

[8] Webopedia. (2010). Programming Language. Retrieved on January 25, 2010 from http://www.webopedia.com/TERM/P/programming_language.html

[9] Voegele, J. (20010). Programming Language Comparison. Retrieved on January 25, 2010 from http://www.jvoegele.com/software/langcomp.html