Auburn University MCSE Design Project:  
Compare and Contrast the Capabilities and Limitations of HTML, Java and VRML  

Summer 1996  

Submitted by  
Barbara M. King  

Submitted to Committee Members  
Dr. W. Homer Carlisle (Chair)  
Dr. James Cross  
Dr. Jennifer McManis  

Technical Report Number: 96-08
## CONTENTS

LIST OF TABLES ................................................................. iii

LIST OF FIGURES ............................................................... iii

ACRONYMS ................................................................ iv

1.0 INTRODUCTION .............................................................. 1

2.0 HTML ........................................................................... 3
  2.1 Introduction ............................................................... 3
  2.2 Language Specifications ............................................... 4
    2.2.1 Document Tags ..................................................... 4
    2.2.2 Basic Text Structure .............................................. 6
    2.2.3 Special Effect Tags ............................................... 7
    2.2.4 Anchors .............................................................. 7
    2.2.5 Images .............................................................. 7
    2.2.6 Special Escape Sequence Characters ....................... 8
  2.3 Enhancements ............................................................ 8
    2.3.1 HTML, version 3.2 ............................................... 9
  2.4 Sample HTML document ............................................. 9

3.0 JAVA ............................................................................. 13
  3.1 Introduction ............................................................. 13
    3.1.1 Java Programs .................................................... 13
    3.1.2 JavaScript ........................................................ 15
    3.1.3 Java Virtual Machine ......................................... 16
  3.2 Language Specifications ............................................... 17
    3.2.1 Data Types and Objects ....................................... 19
    3.2.2 Memory Management and Garbage Collection .......... 20
      3.2.2.1 Java Threads ............................................. 21
    3.2.3 Objects .......................................................... 21
    3.2.4 Classes .......................................................... 21
      3.2.4.1 Constructors ........................................... 22
  3.3 Security Mechanisms .................................................. 22
  3.4 New Developments ..................................................... 23
  3.5 Future Enhancements ................................................ 24
  3.6 Sample Java Applet ................................................... 24
  3.7 Real World Applications ............................................. 28
4.0 VRML ................................................................. 29
  4.1 Introduction ....................................................... 29
    4.1.1 VRML browsers ........................................... 29
  4.2 Language Specifications ......................................... 31
    4.2.1 Separators ............................................... 31
    4.2.2 Nodes .................................................... 31
    4.2.3 General Syntax ........................................... 35
    4.2.4 Fields .................................................... 35
    4.2.5 Point of View ............................................ 36
    4.2.6 Viewing a VRML scene .................................. 36
  4.3 Sample VRML scene ............................................. 37
  4.4 Future Enhancements ............................................ 38
  4.5 Real World Applications ....................................... 39

5.0 CONCLUSIONS ...................................................... 40

6.0 REFERENCES ......................................................... 43

7.0 TERMINOLOGY/GLOSSARY ......................................... 46

APPENDIX A .............................................................. 48
LIST OF TABLES

Table 2-1. HTML Tags .......................................................... 4
Table 2-2. Special Escape Sequence Characters ................................... 8
Table 3-1. Primitive data types and definitions ...................................... 19
Table 3-2. Sizes of data types ...................................................... 20
Table 4-1. Defined nodes ............................................................ 32
Table 4-2. Defined single value fields ............................................. 35
Table 4-3. Defined multiple value fields ......................................... 36

LIST OF FIGURES

Fig. 2-1. HTML request/response flow ............................................ 3
Fig. 2-2. HTML Sample Output ..................................................... 12
Fig. 3-1. Bytecode Verifier ........................................................... 17
Fig. 3-2. Depiction of Java applet ................................................... 27
Fig. 4-1. Depiction of a VRML browser .......................................... 30
<table>
<thead>
<tr>
<th>ACRONYMS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ascii, ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
</tr>
<tr>
<td>CGI</td>
<td>Common Gateway Interface</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name Service</td>
</tr>
<tr>
<td>DTD</td>
<td>Document type definition</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GIF</td>
<td>Graphic Interchange Format</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>JPEG</td>
<td>Joint Photographic Experts Group</td>
</tr>
<tr>
<td>JVM</td>
<td>Java Virtual Machine</td>
</tr>
<tr>
<td>MIDI</td>
<td>Musical Instrument Digital Interface</td>
</tr>
<tr>
<td>MIME</td>
<td>Multimedia Internet Mail Extensions</td>
</tr>
<tr>
<td>NNTP</td>
<td>Network News Transport Protocol</td>
</tr>
<tr>
<td>POSIX</td>
<td>Portable Open Systems Interface</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, green, blue</td>
</tr>
<tr>
<td>SGML</td>
<td>Standard Generalized Markup Language</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>VRML</td>
<td>Virtual Reality Modeling Language</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
<tr>
<td>W3</td>
<td>World Wide Web</td>
</tr>
<tr>
<td>3-D</td>
<td>Three-dimensional</td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

The World Wide Web (WWW, W3, or Web) project, started by CERN (the European Laboratory for Particle Physics), seeks to build a globally distributed hypermedia system. The initial work (1989) focused on the development of the HyperText Transmission Protocol (HTTP), a network protocol for requesting and transmitting WWW files. The WWW is a network of computers providing information via menus to access different resources over the Internet. The WWW provides a uniform interface to services of varying degrees of complexity.

Client programs are referred to as “browsers.” To access the WWW, a user runs a browser program. A browser is a software package which requests information from a server and processes the response to display it in regular document textual form (with graphics, if applicable.)

The WWW is based on a flexible model that allows cross-references (i.e., links or tags) between associated resources. It is based on a technology called hypertext. Hypertext is a method for presenting information in layers (i.e., a way of organizing resources.) A generic, or initial, state (page) is presented with selected words in the text capable of expansion (recognizable by being displayed in a different color than the pure text.) These expansions are links (pointers) to other documents comprised of text files, graphics, or data files. Their purpose is to provide further detail. Expansion may also be achieved through the use of a Common Gateway Interface (CGI) program on the server-side which interprets the form’s data from a client and acts upon it. The CGI specification describes what CGI programs can expect from the standard input, what to send to the standard output, what environment variables to use, and what is allowed on the command line.

HyperText Markup Language (HTML) is a platform-independent language based on the Standard Generalized Markup Language (SGML) used to create hypertext documents. An HTML document can be written with any text editor. HTML is still evolving. Currently available is version 2.0 with version 3.02 due to be released this summer (1996). HTML documents may be retrieved from machines that are running the HTTP daemon.

Hypermedia is a subset of hypertext which allows pointers to other media. This means that browsers might display or produce images, sound, or animations rather than a text file (based on the capabilities/limitations of the client.) A simple text-based browser may display text in paragraphs and lists whereas a more powerful browser may display images and text styles.

The Web uses Uniform Resource Locators (URLs) to achieve the document link to files on other servers. The URL is in the following format:

```
scheme://host.domain[:port]/path/filename
```

The scheme represents a protocol and can be: file, http, gopher, WAIS, news, or telnet; the port number is generally omitted since port 80 serves as the default port for most servers.

An example of this is reading newsletters on the WWW. Each posting contains references to other articles, current or previous, or messages. A client process restructures the postings as hypertext (the
cross-references are turned into links to the references.) This allows the reader to easily move between the original postings and the references by selecting which links to expand (i.e., which path to follow.)

However, as the WWW grows, managers of Web sites continue to look for easier and less complex and less effort-intensive ways to build and maintain online pages. This has resulted in several new Web developments. Along with the growth in the number of active users, sites, and network traffic, the Web’s presentation capabilities are progressing. Web applications have come a long way.

Java is an object oriented programming language and supporting runtime environment developed by Sun Microsystems which allows WWW pages to contain code that is executed on the browser. Java is based on a single “virtual machine.” This enables Java programs to run on any system which has a version of Java. Java was designed to meet the challenges of application development in heterogenous network-wide distributed environments. These challenges include: secure delivery of applications, minimum system resource consumption, hardware and software platform independence, and the capability of performing dynamic extension. HotJava is a WWW browser written in the Java language. It brings interactivity and animation that was previously impossible to client applications on the Web.

Virtual Reality Modeling Language (VRML) is a 3-D scene description language. VRML is one of the latest in a constant series of improvements for the Web. It enhances and broadens the base of possibilities for the future of the Web. VRML adds perceptualization capabilities to the Web; it is a method for publishing and viewing 3-D Web contents. The VRML language is specifically designed for computer graphics. Several browsers are available for multiple platforms. There are a growing number of VRML-based sites on the WWW.

The goal of this research project is to produce a document that compares and contrasts the capabilities and limitations of HTML, Java and VRML. This is discussed in the following sections.

(NOTE: all web site references mentioned in this paper are subject to going offline or relocating without the notice of the author.)
2.0 HTML

HyperText Markup Language (HTML) is a platform-independent language based on the Standard Generalized Markup Language (SGML) used to create hypertext documents. HTML is a collection of styles that define the various components (from a presentation standpoint) of a Web document. An HTML document can be written with any text editor.

2.1 Introduction

HTML provides hypertext links on which the user may select. The browser sends a request across the network to fetch the data associated with that specific link, downloads the data, and displays it locally on the user’s screen. HTTP typically behaves in one of two ways: (1) it gets the document directly; (2) it invokes a CGI program. Refer to Fig. 2-1 for a depiction of the flow.

![Fig. 2-1. HTML request/response flow.](image-url)
2.2 Language Specifications

An HTML document is a collection of styles indicated by the defined markup tags. HTML relies on the markup tags for formatting instructions. The document tags are defined and explained in the following sections.

2.2.1 Document Tags

HTML is composed of tags which are always enclosed within the angle brackets (< >). These tags typically occur in begin-end pairs in the form of `<begin tag> . . . </end tag>`. An exception to the pairing rule is the line break tag. These pairs define the containers (i.e., any content within a container has the rules of that container applied to it; the instructions that tell the browser how to display the various sections of a document.) Examples for containers are boldface and paragraphs. An HTML document always begins and ends with the same tag pair: `<HTML> . . . </HTML>`. Refer to Table 2-1 for a listing of the standard HTML tags, their appearance type (singleton or pair), and their associated definition. HTML is not case sensitive.1

<table>
<thead>
<tr>
<th>Tag</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>Pair</td>
<td>tags that tell a Web browser where the HTML in the document begins and ends</td>
</tr>
<tr>
<td>HEAD</td>
<td>Pair</td>
<td>container for information about the document</td>
</tr>
<tr>
<td>BODY</td>
<td>Pair</td>
<td>contains all of the text and other material that is to be displayed</td>
</tr>
<tr>
<td>TITLE</td>
<td>Pair</td>
<td>contained in the HEAD structure; displayed separately from the text and is usually in a restricted space</td>
</tr>
<tr>
<td>H1 through H6</td>
<td>Pair</td>
<td>levels for headings</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Tag</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM</td>
<td>Pair</td>
<td>emphasis; logical description of the enclosed text -- recommended that emphasis be italicized</td>
</tr>
<tr>
<td>UL</td>
<td>Pair</td>
<td>unordered list</td>
</tr>
<tr>
<td>BR</td>
<td>Singleton</td>
<td>line break</td>
</tr>
<tr>
<td>LI</td>
<td>Singleton</td>
<td>list item</td>
</tr>
<tr>
<td>HR</td>
<td>Singleton</td>
<td>horizontal rule; draws a horizontal line completely across the screen</td>
</tr>
<tr>
<td>STRONG</td>
<td>Pair</td>
<td>strong emphasis; logical description of the enclosed text -- recommended that strong emphasis be bold</td>
</tr>
<tr>
<td>CODE</td>
<td>Pair</td>
<td>computer code</td>
</tr>
<tr>
<td>KBD</td>
<td>Pair</td>
<td>keyboard input</td>
</tr>
<tr>
<td>VAR</td>
<td>Pair</td>
<td>variable name -- the user is to replace the variable with a specific instance</td>
</tr>
<tr>
<td>B</td>
<td>Pair</td>
<td>boldface</td>
</tr>
<tr>
<td>DFN</td>
<td>Pair</td>
<td>for a word being defined -- typically displayed in italics</td>
</tr>
<tr>
<td>CITE</td>
<td>Pair</td>
<td>for titles (books, films, articles, etc.) -- typically displayed in italics</td>
</tr>
<tr>
<td>SAMP</td>
<td>Pair</td>
<td>computer status messages</td>
</tr>
<tr>
<td>I</td>
<td>Pair</td>
<td>italics</td>
</tr>
<tr>
<td>TT</td>
<td>Pair</td>
<td>fixed-width typewriter font</td>
</tr>
<tr>
<td>P</td>
<td>Singleton</td>
<td>new paragraph</td>
</tr>
<tr>
<td>OL</td>
<td>Pair</td>
<td>ordered list</td>
</tr>
<tr>
<td>MENU</td>
<td>Pair</td>
<td>menu list element</td>
</tr>
</tbody>
</table>
### Tag Type Definition

<table>
<thead>
<tr>
<th>Tag</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIR</td>
<td>Pair</td>
<td>directory list element</td>
</tr>
<tr>
<td>FORM</td>
<td>Pair</td>
<td>fill-in forms</td>
</tr>
<tr>
<td>PRE</td>
<td>Pair</td>
<td>preformatted text in fixed font; inclusion of spaces, new lines, and tabs are significant</td>
</tr>
<tr>
<td>BLOCKQUOTE</td>
<td>Pair</td>
<td>quotation tag</td>
</tr>
<tr>
<td>IMG</td>
<td>Singleton</td>
<td>image</td>
</tr>
<tr>
<td>ALIGN</td>
<td>Singleton</td>
<td>alignment of the text or image</td>
</tr>
<tr>
<td>ALT</td>
<td>Singleton</td>
<td>specifies a text alternative for an image</td>
</tr>
<tr>
<td>A</td>
<td>Pair</td>
<td>anchor; the area between the A tags become a hot part of the text</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>Pair</td>
<td>address</td>
</tr>
<tr>
<td>&lt;!--</td>
<td>Pair</td>
<td>comment</td>
</tr>
</tbody>
</table>

#### 2.2.2 Basic Text Structure

Heading structures are used to individually position document or section titles. There are six levels of headings, with H1 (heading, level 1) being the most important or prominently displayed. By default, the headings are displayed in the same font with decreasing point size. A heading always begins at the margin and always forces a line break.

Paragraphs are one of the most basic HTML structures. Web browsers format paragraphs to have one blank line between words regardless of the original structure.

Line breaks are used to force a new line to appear at the place where they are in the text. They can be embedded in paragraphs.

Blockquotes are used for text which is quoted material that requires being set apart and indented from the rest of the text.

There are three types of lists: unordered lists (a list of items where each list element is preceded by a distinctive character), ordered lists (like unordered lists except the Web browser automatically generates a sequence of item markers), and definition lists (based on term-definition pairs rather than list items.)
2.2.3 Special Effect Tags

Special effect tags are used for accenting the text. Logical tags do not directly specify the type of accenting they will use. There are defaults written in the HTML specification, but there is no direct rule. The emphasis and citation tags typically display text in italics; the strong emphasis tag typically displays text in bold.\(^2\)

Forced style tags force a certain style within the document. The most commonly used are: boldface, italics, underline, and typewriter text. Typewriter text is typically used for variable names or for source code examples.\(^3\)

Horizontal rules produce a horizontal line from the left to the right margin. It is commonly used in conjunction with headings for sectional dividers.

2.2.4 Anchors

The anchor tags are always used in conjunction with other forms of tags. For example, when used in conjunction with HREF (hypertext reference), a hyperlink is formed to another resource. When the user activates the link, he/she causes the browser to load the referenced resource. These links are recognizable by text color change and/or underlined. For example, `<A HREF="http://www.auburn.edu/">Auburn University Home Page</A>` takes the user to the Auburn University Home Page.

By using the NAME attribute one can mark points of a document as places that can be jumped to directly rather than having to scroll the entire document to that point. This is most commonly done in the table of contents to take the reader to the specific section of interest. For example, `<p>For clarity, refer to the <a href="#defs">definitions section</a> ... will take the user to ...<p><a name="defs">Definitions</a>`.

2.2.5 Images

The image tag is used in conjunction with the SRC (source) attribute. The source specifies the name and location of the image to incorporate into the document. Wherever the image tag occurs in the document the browser attempts to put the image. It is recommended that the web page developer has all referenced images on his/her PC rather than reference links to them. For example, the line `<img src="hourglass.gif">` causes the icon named “hourglass.gif” to be placed in the text. It should be noted that each instance of an image takes time to process and thus slows down the initial display of the document.

\(^2\) “An Introduction to HTML”, http://www.cru.edu/help/introHTML/TCH1.html

\(^3\) Ibid.
The ALIGN attribute is used in conjunction with the image tag to align the text and image that appear on the same textual line in the document. For example, `<dt><img align=center src="hourglass.gif"><b>TIME</b></dt>` will have the word “TIME” appear next to the image, centered.

The ALT attribute is used to define alternate text for an image in the event that the user’s browser is text-only or if the image loading is turned off. For example, `<img src="help_me.gif" alt="<help me !!!!!!>""> will display `<help me !!!!!!>`.4

### 2.2.6 Special Escape Sequence Characters

There are four ASCII characters that have special meaning in HTML and cannot be used in the text of a document; instead, their escape sequence characters have to be used. Refer to Table 2-2 for a listing.4

#### Table 2-2. Special Escape Sequence Characters.

<table>
<thead>
<tr>
<th>ASCII character</th>
<th>Escape character sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;</code></td>
<td><code>&amp;lt</code></td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td><code>&amp;gt</code></td>
</tr>
<tr>
<td><code>&amp;</code></td>
<td><code>&amp;amp</code></td>
</tr>
<tr>
<td><code>&quot;</code></td>
<td><code>&amp;quot</code></td>
</tr>
</tbody>
</table>

These special escape sequence characters are case sensitive.

### 2.3 Enhancements

Future enhancements for HTML include (from version 1.0 to version 2.0 to version 3.0):

- text color control
- page backgrounds
- tables
- in-line JPEG images
- mathematical characters and functions

---

4 “A Beginner’s Guide to HTML”
2.3.1 HTML, version 3.2

HTML version 3.0 was a proposal for extending HTML version 2.0. The difference between HTML 2.0 and HTML 3.0 was so large that standardization and deployment of the whole proposal proved unwieldy. The HTML 3.0 draft has expired and is not being maintained.\(^5\)

HTML version 3.2 adds the following features to HTML version 2.0: tables, applets, text flow around images, superscripts and subscripts.

Additional features include:\(^6\)

- `<DOCTYPE>` - declaration used to determine HTML version 3.2

- **HEAD** elements:
  - `ISINDEX` - for simple keyword searches
  - `BASE` - defines absolute URL for resolving relative URLs
  - `STYLE` - reserved for future use with style sheets
  - `SCRIPT` - reserved for future use with scripting languages
  - `META` - used to supply meta information as name/value pairs
  - `LINK` - used to define relationships with other documents

- **BODY** elements:
  - The key attributes are `BACKGROUND`, `BGCOLOR`, `TEXT`, `LINK`, `VLINK`, and `ALINK`. These can be used to set a repeating background image plus background and foreground colors for normal text and hypertext links.

2.4 Sample HTML document

The following is a sample HTML document. It was generated by using Microsoft’s FrontPage Editor.

```html
<!DOCTYPE HTML PUBLIC "-//W3O/DTD HTML//EN">
<html>
<head>
<title>barb2</title>
<meta name="FORMATTER" content="Microsoft FrontPage 1.1">
</head>
<frameset cols="35%,65%">
```

\(^5\) “Introducing HTML 3.2”, [http://www.w3.org/pub/WWW/MarkUp/Wilbur/features.html](http://www.w3.org/pub/WWW/MarkUp/Wilbur/features.html)

\(^6\) Ibid.
<frame src="frcont01.htm" name="contents" marginwidth="1" marginheight="1">
<frame src="frmain01.htm" name="main" marginwidth="1" marginheight="1">
<noframes>
<body>
<p>This web page uses frames, but your browser doesn't support them.</p>
</body>
</noframes>
</frameset>
</html>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML//EN">
<html>
<head>
<title>Table of Contents Frame in barb2</title>
<meta name="GENERATOR" content="Microsoft FrontPage 1.1">
<base target="main">
<meta name="FORMATTER" content="Microsoft FrontPage 1.1">
</head>
<body>
<h2 align="center"><img src="images/knife_fork.gif" align="bottom" width="64" height="64"></h2>
</body>
</html>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML//EN">
<html>
<head>
<title>Main Frame in barb2</title>
<meta name="GENERATOR" content="Microsoft FrontPage 1.1">
<meta name="FORMATTER" content="Microsoft FrontPage 1.1">
</head>
<body>
<h2 align="center"><em><img src="images/chili.gif" align="bottom" width="45" height="32">Recipe of the Week Club</em></h2>
</body>
</html>
This week's recipe is for a Western Omelet -- it is appropriate to serve at any meal. Preparation time is approximately 15 minutes. It is great when you aren't really hungry for anything special or if you are in a hurry. Okay, let's get started . . .

Ingredients:

- 3 eggs
- 3 tablespoons 2% milk
- pinch of black pepper
- 1/4 cup cheddar cheese
- 1/4 cup chopped ham
- 2 tbls chopped onions
- 2 tbls chopped green peppers
- 2 tbls sliced fresh mushrooms

Preparation:

Preheat coated skillet to medium heat. Combine beaten eggs, milk and pepper. Mix well. Pour into heated skillet. When mixture starts to bubble, add remaining ingredients. After 3 minutes, carefully fold mixture in half. After 3 minutes, flip. After 3 additional minutes, carefully remove from skillet. Garnish with parsley. Serve with fresh fruit. Serves one.

Comments: Please let me know how you liked:

Last Week's Recipe:

This Week's Recipe:

What would you like to see next week:

Thank you!!!
Recipe of the Week Club

Welcome!

This week's recipe is for a Western Omelet -- it is appropriate to serve at any meal. Preparation time is approximately 15 minutes. It is great when you aren't really hungry for anything special or if you are in a hurry. Okay, let's get started . . .

Ingredients: 3 eggs, 3 tablespoons 2% milk, pinch of black pepper, 1/4 cup cheddar cheese, 1/4 cup chopped ham, 2 tbsls chopped onions, 2 tbsls chopped green peppers, 2 tbsls sliced fresh mushrooms

Preparation: Preheat coated skillet to medium heat. Combine beaten eggs, milk and pepper. Mix well. Pour into heated skillet. When mixture starts to bubble, add remaining ingredients. After 3 minutes, carefully fold mixture in half. After 3 minutes, flip. After 3 additional minutes, carefully remove from skillet. Garnish with parsley. Serve with fresh fruit. Serves one.

Comments: Please let me know how you liked:

Last Week's Recipe:

☐ Yes ☐ No

This Week's Recipe:

☐ Yes ☐ No

What would you like to see next week:

☐ Breakfast ☐ Lunch ☐ Dinner ☐ Dessert

Thank you !!!

Fig. 2-2. Sample HTML output.
3.0 JAVA

Java originated as part of a research project to develop advanced software for a wide variety of networked devices and embedded systems. The goal was to develop a small, reliable, portable, distributed, real-time operating environment. The Java language provides the ability to incorporate video and animation. Java is designed specifically for the Internet.

3.1 Introduction

Java is an object oriented programming language and supporting runtime environment developed by Sun Microsystems, Inc. which allows WWW pages to contain code that is executed on the browser. Java is based on a single “virtual machine” concept. This is an attempt to provide an abstract specification to which builders can design their interpreter without forcing a specific implementation while ensuring that all programs written in Java will be executable on any system that follows the design. This enables Java programs to run on any system which has a version of Java. Java creates a “machine-within-a-machine” which can execute the same programs regardless of the computer’s actual operating system.

Java was designed to meet the challenges of application development in heterogenous network-wide distributed environments. These challenges include: secure delivery of applications, minimum system resource consumption, hardware and software platform independence, and the capability of performing dynamic extension. Java offers special features to allow programs to take advantage of the flexibility of the Internet.

3.1.1 Java Programs

To run Java programs, the user must have the Java run-time systems installed on their platform. This includes a Java interpreter and Java-capable browser. Java-capable browsers include Netscape Navigator 2.0 and HotJava. The Java interpreter is used to execute stand alone Java applications.

There are four different types of programs that can be written using Java: applications, applets, content handlers, and protocol handlers.

Java applications are standalone programs that require the use of the Java interpreter to run. The HotJava Web browser is an example.

Java applets are small applications that are embedded, or included, in Web/HTML pages much like images. Applets are more security conscious and have more behavioral restrictions than the Java applications. A Java applet requires a browser or another Java application to run. Providing the user has a java-capable browser, examples can be viewed at http://www.javasoft.com/applets/applets.htm/. When a Java-capable browser is used to view an HTML page that contains a Java applet, the applet’s code is transferred to the viewer’s system and executed by the browser.\footnote{Ibid.}

The document type definition (DTD) for an applet is as follows:\footnote{Ibid.}

```xml
<!ELEMENT APPLET - - (PARAM*, (%text;)*)>  
<!ATTLIST APPLET 
  CODEBASE CDATA #IMPLIED  -- code base -- 
  CODE CDATA #REQUIRED     -- code file -- 
  ALT CDATA #OPTIONAL     -- alternate text -- 
  NAME CDATA #IMPLIED      -- applet name -- 
  WIDTH NUMBER #REQUIRED 
  HEIGHT NUMBER #REQUIRED 
  ALIGN (left|right|top|texttop|middle| 
       absmiddle|baseline|bottom|absbottom) baseline 
  VSPACE NUMBER #IMPLIED 
  HSPACE NUMBER #IMPLIED 
>
<!ELEMENT PARAM - O EMPTY> 
<!ATTLIST PARAM 
  NAME NAME #REQUIRED     -- The name of the parameter -- 
  VALUE CDATA #IMPLIED    -- The value of the parameter -- 
>
```

Applet resources (including their classes) are normally loaded relative to the document-URL (or <base> tag if it is defined). The codebase attribute is used to change this default behavior. If the codebase attribute is defined then it specifies a different location to find applet resources. The value can be an absolute URL or a relative URL. The absolute URL is used as is without modification and is not effected by the documents <base> tag. When the codebase attribute is relative, then it is relative to the document-URL (or <base> tag if defined).\footnote{Ibid.}
Here is an example:11

```html
<applet codebase="applets/NervousText"
    code=NervousText.class
    width=300
    height=50>
    <param name=text value="Java is Cool!">
</applet>
```

Java content handlers and Java protocol handlers are Java programs/mechanisms that allow Java Web browsers to dynamically understand and handle new data types. Java comes with several content and protocol handlers for GIFs, bitmaps, FTP, and the network news transport protocol (NNTP).12

### 3.1.2 JavaScript

Sun Microsystems, Inc. and Netscape Communications Corp. have announced an open, cross-platform object scripting language for the creation and customization of applications on networks and the Internet. JavaScript is an object scripting language designed for creating live online applications that link together objects and resources on both clients and servers. The targeted audience is HTML page authors and application developers. On the server, the scripts might compose and format HTML content based on user preferences stored in a relational database; on the client, the scripts would glue together an assortment of Java applets and HTML form elements into a live interactive user interface for specifying a netwide search for information.13

For example:14

```html
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!-- to hide script contents from old browsers

    function square(I) {
        document.write("The call passed ", I, " to the function.")
        return I * I
    }

    document.write("The function returned ",square(5),".")
// end hiding contents from old browsers -->
</SCRIPT>
</HEAD>
```

11 Ibid.

12 Java!

13 “Sun, Netscape introduce JavaScript object scripting language”, The Sun Observer, p. 6, February 1996.

14 “Navigator Scripting”, http://home.netscape.com/eng/mozilla/Gold/handbook/javascript/script.html#C1
All done.

JavaScript is said to complement the Java language and the initial version is available as part of the Netscape Navigator 2.0 beta release.\(^{15}\)

### 3.1.3 Java Virtual Machine

The Java Virtual Machine (JVM) provides the hardware platform specification to which all Java code is compiled. JVM consists of the following five specifications which control the implementation and interpretation of Java code:\(^{16}\)

1. **The Instruction Set** - Each instruction in Java consists of an opcode followed by an optional operand. JVM uses big endian encoding scheme.

2. **The Register Set** - Hold information that the processor uses to store the current state of the system. Each register is 32 bits. The register set is limited to the following four registers:
   - *pc* - program counter
   - *optop* - pointer to the top of the operand stack
   - *frame* - pointer to the current execution environment
   - *vars* - pointer to the first local variable of the current execution environment

3. **The Operand Stack** - The operand stack is first-in-first-out 32 bit size stack that holds the arguments necessary for the opcodes. It is the primary area for storage of current status information.

4. **The Garbage Collected Heap** - The job of memory deallocation and garbage collection is the responsibility of the runtime environment.

---

\(^{15}\) Ibid.

\(^{16}\) Java\(^{1}\), p.336 - 342

16
3.2 Language Specifications

Java provides extensive compile-time checking followed by a second level of run-time checking. The Java compiler does not compile references down to numeric values. Instead, it passes symbolic reference information through to the bytecode verifier and the interpreter. The bytecode verifier traverses the bytecodes, constructs the type state information, and verifies the types of the parameters to all the bytecode instructions. Refer to Fig. 3-1 for a depiction.\(^\text{17}\)

---

The figure illustrates the flow of data and control from Java language source code through the Java compiler, to the bytecode verifier, class loader, and to the Java interpreter. The Java bytecode loader and the bytecode verifier ensure that code passed on to the Java interpreter is in a state to be executed and does not violate system integrity or Java standards. Imported code is not allowed to execute until after it has passed the bytecode verifier testing.

The class loader brings together all the code necessary for application execution including inherited and called classes.

The Java interpreter performs final name resolution once, when classes are being linked. Once the name is resolved, the reference is rewritten as a numeric offset.

The storage layout of objects is not determined by the compiler. The layout of objects in memory is deferred to runtime and determined by the interpreter. Updated classes with new instance variables or methods can be linked in without affecting existing code.\(^\text{18}\)

Java is designed to operate in distributed environments which means that, with the security features designed into the language and run-time system, Java lets you construct applications that can not be invaded from the outside. The Java compiler generates bytecodes, a high-level architecture neutral intermediate format designed to transport code efficiently to multiple hardware and software platforms (similar to machine instructions); the same Java language bytecodes will run on any platform. Java bytecodes are designed to be easy to interpret on any machine or to dynamically translate into native machine code if performance requires.\(^\text{19}\)

However, Java still has its vulnerabilities. In January 1996, three students at Princeton University discovered a way to use Java to exploit a domain name service (DNS) vulnerability. This is not a vulnerability with Java, per se; it is a vulnerability with DNS that Java does not check for. DNS servers provide a means for translating between host names and IP addresses. The problem occurs because Java does not completely verify that the information provided by the DNS server is accurate.\(^\text{20}\)

In February 1996, a student of Oxford University made public a security problem with Java related to the loading of local class libraries. The student discovered a way to load class libraries from any readable directory on a user’s system.\(^\text{21}\)


\(^{20}\) CIAC Notes, March 18, 1996, Number 96-01

\(^{21}\) Ibid.
The architecture-neutral and portable language environment of Java is known as the Java Virtual Machine (JVM). It is the specification of an abstract machine for which Java language compilers can generate code. Specific implementations of the JVM for specific hardware and software platforms then provide the concrete realization of the virtual machine. The JVM is based primarily on the POSIX interface specification -- an industry standard definition of a portable system interface. Implementing the JVM on new architectures is a relatively straightforward task as long as the target platform meets basic requirements such as support for multithreading.\(^{22}\)

The complete Java system includes a number of libraries of utility classes and methods of use to developers in creating multi-platform applications. These libraries are:

- **java.lang** -- the collection of base types that are always imported into any given compilation unit
- **java.io** -- streams and random-access files
- **java.util** -- container and utility classes
- **java.awt** -- an Abstract Windowing Toolkit (AWT) that provides an abstract layer enabling the user to port Java applications easily from one window system to another

### 3.2.1 Data Types and Objects

Everything in Java is an object; even the primitive data types can be encapsulated inside library-supplied objects. Refer to Table 3-1 for primitive data types and definitions.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric</td>
<td>Integer -- 8-bit byte, 16-bit short, 32-bit int, and 64-bit long</td>
</tr>
<tr>
<td></td>
<td>Real -- 32-bit float and 64-bit double</td>
</tr>
<tr>
<td>Boolean</td>
<td>distinct data type -- true or false</td>
</tr>
<tr>
<td>Arrays</td>
<td>real object with a runtime representation; can declare and allocate arrays of any type and can allocate arrays of arrays to obtain multi-dimensional arrays</td>
</tr>
</tbody>
</table>

\(^{22}\) Ibid.
The data types and sizes referenced in Table 3-2 are standard across all implementations of Java.

Table 3-2. Sizes of data types.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>8-bit two’s complement</td>
</tr>
<tr>
<td>short</td>
<td>16-bit two’s complement</td>
</tr>
<tr>
<td>int</td>
<td>32-bit two’s complement</td>
</tr>
<tr>
<td>long</td>
<td>64-bit two’s complement</td>
</tr>
<tr>
<td>float</td>
<td>32-bit IEEE 754 floating point</td>
</tr>
<tr>
<td>double</td>
<td>64-bit IEEE 754 floating point</td>
</tr>
<tr>
<td>char</td>
<td>16-bit Unicode character</td>
</tr>
</tbody>
</table>

Strings are Java language objects; the `String` class is for read-only objects and the `StringBuffer` class is for modifiable string objects. Java has also extended the meaning of the ‘+’ operator to indicate string concatenation.

3.2.2 Memory Management and Garbage Collection

Java’s memory management model is based on objects and references to objects. Once the user has allocated an object, the runtime system keeps track of the object’s status and automatically reclaims memory when objects are no longer in use. When an object has no more references, the object is a candidate for garbage collection.23

Automatic garbage collection is an integral part of Java and its runtime system. The garbage collector gathers and compacts unused memory increasing the probability that adequate memory resources are available when needed during periods of heavy interactive use. This is achieved by the Java runtime system taking advantage of idle CPU periods and running the garbage collector in a low priority thread.

3.2.2.1 Java Threads

Running a single-threaded program is fine when the user only has to deal with small programs that have to do a single task, but viewing a web page where several applets are supposed to be running at the same time would have to be multi-threaded.\(^{24}\)

Threads are an integral foundation of Java. The Java library provides a `Thread` class that supports a collection of methods to start, run, stop, and check the status of a thread. Java thread support includes a set of synchronization primitives based on the monitor and condition variable paradigm. Java supports multithreading at the language level and from its runtime system and thread objects.\(^{25}\)

In Java, there are 2 ways to create a class that can be multithreading:\(^{26}\)
- create a class that extends the thread class, or
- create a class that implements the runnable interface.

3.2.3 Objects

Object technology is a collection of analysis, design, and programming methodologies that focus design on modeling the characteristics and behaviors of objects in the real world. Objects are software programming models. Objects have state and behavior. An object’s state is defined by its instance variable (private to the object); an object’s behavior is defined by its methods (manipulate the instance variables.)\(^{27}\)

With the exception of the primitive data types (numbers and booleans), everything in Java is an object; however, even the primitive types have object counterparts or can be encapsulated within objects.

Objects communicate by sending messages to each other and respond to messages by selecting a method to execute.

3.2.4 Classes

A class is not an object; rather, it is a template that defines how an object will look and behave when the object is created or instantiated from the specification declared by the class. A class defines the instance variables and methods of an object. Many objects can be instantiated from one class.

\(^{24}\) Java!

\(^{25}\) Ibid.

\(^{26}\) Ibid.

\(^{27}\) “The Java Language Environment: A White Paper”
definition. In other words, Java code is organized into classes. Each class defines a set of methods that define the behavior of an object. A class can inherit behaviors from another class. At the root of the class hierarchy is always class Object.

Java supports a single-inheritance class hierarchy. This means that each class can only inherit from one other class at a time.

Subclassing and containment are the means by which new, and possibly enhanced, objects can be defined in terms of existing objects.

3.2.4.1 Constructors

In Java, when a class is declared, the user can also declare optional constructors that perform initialization when objects are instantiated from that class. There are situations when constructors are necessary, i.e., when the object being instantiated must itself in turn instantiate other objects or initialize itself by means of code execution.

For example, the thread constructors create an instance of the thread class, allowing the object to be run concurrently with other objects. When creating a thread, one can either create the thread by subclassing Thread itself; or, if one has a class that implements Runnable, pass that class to the thread constructor, which will then create a thread out of that class.

3.3 Security Mechanisms

The networking package in Java provides the interfaces to handle various network protocols such as FTP, HTTP, Telnet, etc. The networking package can be configured to:

- disallow all network access
- allow network access only to hosts from which code was imported
- allow network access outside the firewall only if code came from outside
- allow all network access

---

28 Ibid.

29 “What is This Thing called Java?”, Authur van Hoff, Sami Shaio, Orca Starbuck, Datamation, p.45-46, March 1, 1996.

30 Ibid.


32 Java!
Java’s security mechanisms work at four different levels of the system architecture:\textsuperscript{33}, \textsuperscript{34}

- First: The Java language and its compiler are the first line of defense. The Java language itself was designed to be safe, and the Java compiler ensures that source code does not violate the rules. This pass ensures that the class file has the format of a class file.

- Second: All bytecodes executed by the Java runtime engine are screened to be sure that they also obey these rules. This layer guards against having an altered compiler produce code that violates the safety rules. Before running any bytecodes, the runtime subjects them to a rigorous series of tests that vary in complexity from simple format checks to running a theorem prover. The verifier performs all verification that can be performed without looking at the bytecodes.

- Third: This is the most complex pass of the class verification. The class loader ensures that classes do not violate name space or access restrictions when they are loaded into the system. The class loader never allows a class from a less protected realm to replace a class from a more protected realm. In addition, classes in one realm cannot call upon the methods of classes in other realms unless those classes have explicitly declared those methods public.

- Fourth: Application Program Interface-specific (API) security prevents applets from doing destructive things. This layer depends on the security and integrity guarantees from the other three layers.

3.4 New Developments

Java possibilities are growing daily. JavaSoft, an operating company of Sun Microsystems Inc., announced JDBC, a database access application programming interface (API) that enables developers to write Java applications that access databases. This API extends the Web by allowing embedded Java applications to access stored data in conventional databases.\textsuperscript{35}

\textsuperscript{33} “Yes, Java’s Secure. Here’s Why”, Laura Lemay, Charles Perkins, Datamation, p.47-49, March 1, 1996.


\textsuperscript{35} “Java to integrate with enterprise databases”, The Sun Observer, p.4, April 1996.
3.5 Future Enhancements

Future enhancements for Java include:36

- development of a compiler to let developers compile portions of a Java applet for specific operating systems for performance improvement
- an object linking and embedding control extension within the Java engine
- integrate Java with common object request broker architecture compliant back end databases
- support other programming languages with Java
- new multimedia and graphics classes

3.6 Sample Java Applet

The following code “animates” and changes the color of the literal string “Let’s Eat !!!”.

/* This applet is a combination with modifications of what was originally written by Daniel Wyszynski (NervousText.java) and Arthur van Hoff (Blink.java)
file: Jitters1.java, Barb King */

import java.awt.*;
import java.awt.image.*;

public class Jitters1 extends java.applet.Applet implements Runnable {
    char separated[];
    String s1 = null;
    String s2 = null;
    Thread killme = null;
    int i, x_coord = 0, y_coord = 0;
    String num;
    int speed=35, counter =0;
    Image image;
    boolean threadSuspended = false;
    Font font;

36 Java!
public void init() {
    s1 = getParameter("text");
    if (s1 == null) {
        s1 = "Let’s Eat !!!");
    }
    s2 = getParameter("img");
    if (s2 == null) {
        s2 = "chili.gif";
    }
    separated = new char [s1.length()];
    s1.getChars(0,s1.length(), separated, 0);
    image = getImage(getCodeBase(),"chili.gif");
    resize(150,50);
    font = new java.awt.Font("TimesRoman", Font.PLAIN, 36);
}

public void start() {
    if (killme == null) {
        killme = new Thread(this);
        killme.start();
    }
}

public void stop() {
    killme = null;
}

public void run() {
    while (killme != null) {
        try {Thread.sleep(100);} catch (InterruptedException e){}
        repaint();
    }
    killme = null;
}

public void paint(Graphics g) {
    int x = 0, y = font.getSize();
    g.drawImage(image,75,75,this);
    int red = (int)(Math.random() * 50);
    int green = (int)(Math.random() * 50);
    int blue = (int)(Math.random() * 256);
    g.setColor(Color.black);
    g.setFont(font);
    for(i=0;i<s1.length();i++){
        x_coord = (int) (Math.random()*10+15*i);
        y_coord = (int) (Math.random()*10+36);
        if (Math.random() < 0.5) {
            g.setColor(new java.awt.Color((red+y*30)%256,
                                         (green+x/3)%256, blue));
        }
    }
}
The following is the HTML file used to run the applet:

```
<hr>
<applet code="Jitters1.class" width=200 height=50>
<param name=text value="Let's Eat !!!">
<param name=img value="chili.gif">
</applet>
<hr>
<a href="Jitters1.java"></a>
<p>This applet is a modification to applets originally written by Daniel Wyszynski (NervousText.java) and Arthur van Hoff (Blink.java)</p>

The applet HTML file was included in the frame HTML file to add animation to the original web page. Refer to Fig. 3-2 for a depiction of a Java applet web page.
Recipe of the Week Club

Welcome!

This week's recipe is for a Western Omelet. It is appropriate to serve at any meal. Preparation time is approximately 15 minutes. It is great when you aren't really hungry for anything special or if you are in a hurry. Okay, let's get started...

Ingredients: 3 eggs, 2 tablespoons 2% milk, pinch of black pepper, 1/4 cup cheddar cheese, 1/4 cup chopped ham, 2 thick chopped onions, 2 thick chopped green peppers, 2 thick sliced fresh mushrooms

Preparation: Preheat coated skillet to medium heat. Combine beaten eggs, milk and pepper. Mix well. Pour into heated skillet. When mixture starts to bubble, add remaining ingredients. After 3 minutes, carefully fold mixture in half. After 3 minutes, flip. After 3 additional minutes, carefully remove from skillet. Garnish with parsley. Serve with fresh fruit. Serve one.

Comments: Please let me know how you liked:

Last Week's Recipe:
- Yes
- No

This Week's Recipe:
- Yes
- No

What would you like to see next week:
- Breakfast
- Lunch
- Dinner
- Dessert

Thank you!!!
3.7 Real World Applications

The following is a short list of real world applications using Java:

- At the Computer Museum in Boston, independent consultant Daniel Griscom is using Java to bring the museum’s Networked Puzzle exhibit to the Internet. The Networked Puzzle problem allows museum visitors to cooperatively assemble a twelve-piece puzzle out of sight of each other.\(^\text{37}\)

- Cadis Inc.’s Krakatoa application lets companies such as National Semiconductor Corporation publish product information over the Web and provide interactive access to up-to-date data. Cadis developed both the client application in Java and the Java remote procedure calls so that information from the National Semiconductor Corporation Web server can be viewed by clients in real-time as it is updated.\(^\text{38}\)

- The Internet Shopping Network has created a Java application designed to allow its subscribers to bid on goods in a real-time interactive auction.\(^\text{39}\)

- The capital investments branch of First Union National Bank has used Java to develop cost-effective portable applications such as an up-to-the-minute market and corporate data delivery system.\(^\text{40}\)

- R. R. Donnelley and Sons Co., a periodical and print publisher in Chicago, turned to Java to create an application that allows customers to access the company’s online catalog and publishing system.\(^\text{41}\)

---

\(^{37}\) “Puzzle me this, puzzle me that”, Rebecca Sykes, February 15, 1996, JavaWorld Online, http://www.javaworld.com/cgi-bin/w3com/start?JW+main


\(^{39}\) Ibid.


4.0 VRML

Virtual Reality Modeling Language (VRML) is one of the latest in a constant series of improvements for the Web. It enhances and broadens the base of possibilities for the future of the Web. VRML adds perceptualization capabilities to the Web; it is a method for publishing and viewing 3-D Web contents. The VRML language is specifically designed for computer graphics. It is a scene description language -- not a programming language. VRML is parsed (transformed into computer understandable objects) and is displayed on the screen. Nothing changes once the VRML file is loaded except the user’s point of reference/view.

4.1 Introduction

HTML is designed for two dimensional text -- not graphics. VRML is a graphical 3-D scene description language. Graphical 3-D, in this context, refers to geometry, transformations, lighting/shading, adding textures to images, and clipping/transformations. VRML is a language for describing multi-user interactive simulations. VRML is essentially a 3-D equivalent of HTML (a document-oriented view of the Web), but adds from a perceptual setting. It is an object-oriented scene description language.\(^\text{42}\)

It is the intention of its designers that VRML become the standard language for interactive simulation within the WWW; in fact, demand for VRML is being driven by its potential for commercial applications.\(^\text{43}\)

4.1.1 VRML browsers

The browser (or viewer) is used to interpret VRML worlds and make them visible on a computer. The browser has several components: a network interface, a VRML parser, a renderer, and a navigation interface.

There are two approaches for a VRML browser to get the data into the browser: the browser can be implemented as a helper application or it can be implemented as a standalone application.


\(^{43}\) “A Whole New Virtual World”, John Vacca, Information Week, p. 1a-4a, Feb 26, 1996.
Helper applications can be used in conjunction with another application to provide additional capabilities. In general, a Web browser application must be running while a VRML helper application is being used.

Standalone applications are fully equipped Web browsers. They do not require the use of other applications to display VRML files. Standalone applications behave as if they are helper applications in the sense that they launch the appropriate browser if necessary to display a retrieved file.

Integrated browsers are those that can understand a variety of data file types. This means that it is possible to click on a link to a VRML world from within an HTML document and, rather than going to a separate application, the application opens a window into the VRML world.

The VRML parser converts the VRML token stream from ASCII format into a form that the computer understands. The VRML browser receives data through its network interface and passes it on to the parser. The parser then “walks” the data within the VRML file and converts this file into an internal representation referred to as a parse tree. This parse tree is then traversed until the entire VRML file has been translated into a visible representation, or rendered on the computer’s display.

The navigation interface implements the six degrees of freedom -- movement in x, y, and z, and orientation in yaw, pitch, and roll. Collision detection is also implemented. Refer to Fig. 4-1 for a depiction of a VRML browser. 

---

Fig. 4-1. Depiction of a VRML browser.

---

44 VRML Browsing & Building Cyberspace
4.2 Language Specifications

At the highest level, VRML is a way for objects to read and write themselves. In theory, the objects can contain anything (3-D geometry, MIDI data, JPEG files, etc.) A VRML scene (also referred to as a “document” or a “world”) has three basic types of information: separators, nodes, and fields. These are defined in the following sections.\(^{45}\)

VRML requires no changes to the way Web servers operate; however, the Web server has to be informed that VRML has a .wrl extension and the Multimedia Internet Mail Extensions (MIME) type is x-world/x-vrml.

All of the same linking that is found in the Web today are also present in VRML. For instance, the user can click on an object and play a movie or hear a sound. An object, or any part of an object, can be linked to any object available through the Web in a VRML world. It is even possible to link VRML worlds together (referred to as “teleporting” from world to world.)

4.2.1 Separators

The role of the separator is to establish the scope. Any field or node (defined later) within a separator have scope only within the range of that specific separator. It is a group node -- it gathers everything within it into a single unit.\(^{46}\) In other words, changes made to the items within a separator do not affect items outside of it.\(^{47}\)

4.2.2 Nodes

VRML defines a set of objects, referred to as nodes, for doing 3-D graphics (i.e., the doing part of VRML; describe the material features of a scene.) Nodes are arranged in hierarchical structures referred to as scene graphs. Scene graphs define an ordering for the nodes and has a notion of state -- earlier nodes can affect later nodes.\(^{48}\) Nodes have the following characteristics: \(^{49}\)

- the kind of object -- cube, sphere, texture map, transformation, etc.


\(^{47}\) VRML Browsing & Building Cyberspace

\(^{48}\) Ibid.

\(^{49}\) “The Virtual Reality Modeling Language Version 1.0 Specification (Draft)”
• the parameters (fields) that distinguish this node from others of the same type -- radius of a sphere object, image of texture map, etc.

• identifying name -- uniquely named nodes for identification purposes to allow reference to them elsewhere

• hierarchical -- object hierarchy is implemented by allowing nodes to contain other nodes (parent and children)

Nodes are classified into one of three categories:

• shape -- define the geometry in the scene
• property -- affect the way shapes are drawn
• group -- gather other nodes together allowing collections of nodes to be treated as a single object

Refer to Table 4-1 for defined nodes.

Table 4-1. Defined nodes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cone</td>
<td>represents a simple cone whose central axis is aligned with the y-axis; default: centered at (0,0,0), size of -1 to +1 in all three directions</td>
</tr>
<tr>
<td>Coordinate3</td>
<td>defines a set of 3-D coordinates to be used by a IndexedFaceSet node</td>
</tr>
<tr>
<td>Cube</td>
<td>represents a cube aligned with the coordinate axes; default: centered at (0,0,0), size of -1 to +1</td>
</tr>
<tr>
<td>Cylinder</td>
<td>represents a simple capped cylinder centered around the y-axis; default: centered at (0,0,0), size of -1 to +1 in all three dimensions</td>
</tr>
<tr>
<td>DirectionalLight</td>
<td>defines a directional light source that illuminates along rays parallel to a given 3-D vector</td>
</tr>
<tr>
<td>Group</td>
<td>defines the base class for all group nodes; contains an ordered list of children nodes</td>
</tr>
<tr>
<td>Name</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IndexedFaceSet</td>
<td>represents a 3-D shape formed by constructing polygons from vertices located at the current coordinates</td>
</tr>
<tr>
<td>IndexedLineSet</td>
<td>represents a 3-D shape formed by constructing polylines from vertices located at the current coordinates</td>
</tr>
<tr>
<td>Info</td>
<td>defines an information node in the scene graph</td>
</tr>
<tr>
<td>LevelOfDetail</td>
<td>used to allow applications to automatically switch between various representations of objects</td>
</tr>
<tr>
<td>Material</td>
<td>defines the current surface material properties for all subsequent shapes</td>
</tr>
<tr>
<td>MaterialBinding</td>
<td>specifies how the current materials are bound to shapes that follow in the scene graph</td>
</tr>
<tr>
<td>MatrixTransform</td>
<td>defines a geometric 3-D transformation with a single SbMatrix</td>
</tr>
<tr>
<td>Normal</td>
<td>defines a set of 3-D surface normal vectors to be used by vertex-based shape nodes that follow it in the scene graph</td>
</tr>
<tr>
<td>NormalBinding</td>
<td>specifies how the current normals are bound to shapes that follow in the scene graph</td>
</tr>
<tr>
<td>OrthographicCamera</td>
<td>defines a parallel projection from a viewpoint</td>
</tr>
<tr>
<td>PerspectiveCamera</td>
<td>defines a perspective projection from a viewpoint</td>
</tr>
<tr>
<td>PointLight</td>
<td>defines a point light source at a fixed 3-D location</td>
</tr>
<tr>
<td>PointSet</td>
<td>represents a set of points located at the current coordinates</td>
</tr>
<tr>
<td>Rotation</td>
<td>defines a 3-D rotation about an arbitrary axis through the origin</td>
</tr>
<tr>
<td>Scale</td>
<td>defines a 3-D scaling about the origin</td>
</tr>
<tr>
<td>Name</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Separator</td>
<td>performs a push (save) of the traversal state before traversing its children and a pop (restore) after traversing them</td>
</tr>
<tr>
<td>ShapeHints</td>
<td>indicates that IndexedFaceSets are solid, contain ordered vertices, or contain convex faces</td>
</tr>
<tr>
<td>Sphere</td>
<td>represents a sphere; default: centered at the origin and has a radius of 1</td>
</tr>
<tr>
<td>SpotLight</td>
<td>defines a spotlight style light source</td>
</tr>
<tr>
<td>Switch</td>
<td>traverses only one or none of its’ children</td>
</tr>
<tr>
<td>Texture2</td>
<td>defines a texture map and parameters for that map</td>
</tr>
<tr>
<td>Texture2Transform</td>
<td>defines a 2-D transformation applied to texture coordinates</td>
</tr>
<tr>
<td>TextureCoordinate2</td>
<td>defines a set of 2-D coordinates to be used to map textures to subsequent vertex-based shapes</td>
</tr>
<tr>
<td>Transform</td>
<td>defines a geometric 3-D transformation consisting of a non-uniform scale about an arbitrary point, a rotation about an arbitrary point and axis, and a translation</td>
</tr>
<tr>
<td>TransformSeparator</td>
<td>saves current transformation state before traversing its children and restores it afterwards</td>
</tr>
<tr>
<td>Translation</td>
<td>defines a translation by a 3-D vector</td>
</tr>
<tr>
<td>WWWAnchor</td>
<td>loads a new scene into a VRML browser when one of its’ children is chosen</td>
</tr>
<tr>
<td>WWWInline</td>
<td>reads its children from anywhere in the Web</td>
</tr>
</tbody>
</table>

Instancing, using the same instance of a node multiple times, is allowed.
4.2.3 General Syntax

For identification purposes, every VRML file begins with the characters `#VRML V1.0 ascii`. Anything after these characters on the same line is ignored. The `#` character begins a comment. All characters until the next newline or carriage return are ignored with string fields being the only exception. After the required header, a VRML file contains exactly one VRML node. That node may be a group node (i.e., nodes that may have children) containing any number of other nodes.

4.2.4 Fields

Two general classes of fields exist: those that contain a single value and those that contain multiple values. Each field type defines the format for the values it writes.

- **single value** -- a value may be a single number, a vector, or an image. All single value fields have names that begin with “SF”. Refer to Table 4-2 for defined single value fields.

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFBitMask</td>
<td>field that contains a mask of bit flags; nodes that use this field class define mnemonic names for the bit flags</td>
</tr>
<tr>
<td>SFBool</td>
<td>field containing a single boolean value</td>
</tr>
<tr>
<td>SFColor</td>
<td>field containing a color</td>
</tr>
<tr>
<td>SFEEnum</td>
<td>field that contains an enumerated type; nodes that use this field class define mnemonic names for the values</td>
</tr>
<tr>
<td>SFFloat</td>
<td>field containing a single precision floating point number</td>
</tr>
<tr>
<td>SFLong</td>
<td>field containing a single long integer (32-bit)</td>
</tr>
<tr>
<td>SFMatrix</td>
<td>field containing a transformation matrix</td>
</tr>
<tr>
<td>SFRotation</td>
<td>field containing an arbitrary rotation</td>
</tr>
<tr>
<td>SFString</td>
<td>field containing an ASCII string</td>
</tr>
<tr>
<td>SFVec2f</td>
<td>field containing a two-dimensional vector</td>
</tr>
<tr>
<td>SFVec3f</td>
<td>field containing a three-dimensional vector</td>
</tr>
</tbody>
</table>
multiple values -- written as a series of values separated by commas enclosed in square brackets. All multiple value fields have names that begin with “MF”. Refer to Table 4-3 for defined multiple value fields.

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFColor</td>
<td>field that contains any number of RGB colors</td>
</tr>
<tr>
<td>MFLong</td>
<td>field that contains any number of long integers (32-bit)</td>
</tr>
<tr>
<td>MFVec2f</td>
<td>field that contains any number of two-dimensional vectors</td>
</tr>
<tr>
<td>MFVec3f</td>
<td>field that contains any number of three-dimensional vectors</td>
</tr>
</tbody>
</table>

4.2.5 Point of View

The point of view (point of reference) of the VRML world presented to the user is through a “camera.” In some viewers, viewpoints are created through menu selections; in other viewers, it is possible to set the viewpoints. Viewpoints are used to allow a user to quickly jump around between or to points of interest in a VRML world.50

4.2.6 Viewing a VRML scene

The steps required for viewing a VRML scene are as follows:

- a document request is sent from a browser
- the Web server that receives the request replies
- the reply is sent to the requesting browser

Once the document has been received, it is parsed (or interpreted.) Visible representations of the objects described in the VRML document are displayed. The interface for navigation and the scene’s camera are activated.51

50 VRML Browsing & Building Cyberspace

51 Ibid.
VRML files are often loaded in stages (due to the ability for distribution by referencing objects at different Web sites.) First, the basic scene description is loaded. If necessary, the nested scenes are then loaded (scenes within the basic scene.)

4.3 Sample VRML scene

The following scene contains a model of a track light consisting of primitive shapes, plus three walls.\(^{52}\)

```vrml
#VRML V1.0 ascii
Separator {
  Separator { #simple track light geometry:
    Translation { translation 0 4 0 }

  }

  Material { emissiveColor 0.1 0.3 0.3 }
  Cube {
    width 0.1
    height 0.1
    depth 4
  }
}

Rotation { rotation 0 1 0 1.57079 }
Separator {
  Material { emissiveColor 0.3 0.3 0.1 }
  Rotation { rotation 1 0 0 1.57079 }
  Translation { translation 0 -.2 0 }
  Cone {
    height .4
    bottomRadius .2
  }
  Translation { translation 0 .4 0 }
  Cylinder {
    radius 0.02
  }
}

SpotLight { #light from above
  location 0 4 0
  direction 0 -1 0
  intensity 0.9
  cutOffAngle 0.7
```

\(^{52}\) VRML Browsing & Building Cyberspace
S
Separator {  #wall geometry; three flat polygons
  Coordinate3 {
    point [
      -2 0 -2, -2 0 2, 2 0 2, 2 0 -2,
      -2 4 -2, -2 4 2, 2 4 2, 2 4 -2
    ]
  }
  IndexedFaceSet {
    coordIndex [ 0, 1, 2, 3, -1,
      0, 4, 5, 1, -1,
      0, 3, 7, 4, -1
  }
}
WWWAnchor {  #hyperlinked object
  name “http://www.foo.edu/CowProject/AboutCows.html”
  Separator {
    Translation { translation 0 1 0 }
    WWWInline {  # reference another object
      name “http://www.foo.edu/3DObjects/cow.wrl”
    }
  }
}

4.4 Future Enhancements

Future enhancements for VRML include (specifications for versions 1.x and 2.0):  

- addition of a facility to define objects as a series of curves
- more mature texture mapping
- dynamic VRML worlds
- modify the behavior of VRML browsers so that individual VRML objects can be sent to a browser from a server in the context of a larger scene
- specify an API for interactivity (i.e., a standard methodology through which interactivity takes place within the VRML browser without specifying how the interactivity occurs)
- add the capability to attach a data stream (i.e., a continuous flow of data; addition of audio and video) to a VRML world

53 VRML Browsing & Building Cyberspace
4.5 Real World Applications

The following is a short list of real world applications using VRML:

- Lightscape (http://www.lightscape.com/models.html) has the following scenes available: Image Gallery, Jerusalem City Hall model, Operating Room model and Laboratory model.  

- VRML is playing a big part in allowing researchers to work with chemical and crystal structure models. Chemists use VRML to visualize molecules before they are chemically synthesized.54

- The construction industry uses VRML to explore exact representations of architectural drawings and models before starting to build.55

- VRML is used extensively in the gaming industry.56

- Visa is developing virtual banking and retailing applications delivered via the Internet. Within virtual rooms, users can interact with other individuals (i.e., bank tellers, loan officers, financial advisers) who can help them conduct business online.57

- The real estate industry plans to capitalize on VRML by showing houses on the Internet to long distance clients.58

- Residents of San Francisco’s SoMa neighborhood are working together to create a VRML-based “Virtual SoMa.” When completed, people will be able to take virtual walking tours of the neighborhood from their computer desktop.59

---


55 Ibid.

56 Ibid.

57 Ibid.

58 Ibid.

59 “3-D technology hits the Web”, Richard Karpinski, Interactive Age, October 1995.
5.0 CONCLUSIONS

1) Currently, HTML cannot control the following items:

- typeface, centering, spacing or line breaks used for any document component other than the preformatted text
- the point-size for any specific font
- the width or height of the screen
- the background, foreground, or highlight colors

Later versions of HTML might accommodate some of these (version 1 to version 2 to version 3).

However, the positive characteristics of HTML include:

- HTML/HTTP is quickly becoming the standard of information servers
- ease of programming
- several easy to use browsers are available
- support for graphics, sound, and movies
- CGI and forms capabilities

Also, due to the amount of HTML sources on the Web, one could easily download the HTML source to use as a template to design a Web page. One could also easily download bitmaps for backgrounds and GIF files for icons. Some vendors, Netscape for example, have combined with their product an HTML editor.

2) A 32-bit operating system is required to drive Java applications. Java contains multithreading capabilities that cannot be easily handled by Windows 3.1. A significant drawback to the interpreted code is that it runs slower than compiled code. Another drawback to Java is the limit, lack of, or immaturity of developing tools. Although the number of tools is increasing dramatically, it appears they need refinement. On the other hand, Java will give an opportunity to visually impaired individuals to navigate the Web by “listening.”

Java, and the technology it represents, promises a totally new way of providing services and products to consumers and an effective way of programming on the Web. It has been predicted that Java will dramatically change the software distribution industry. Also, because Java is platform independent, software developers only need to create and test one version of the application -- there is no need to write or maintain separate versions for the various platforms. Real time execution architectures, like Java, will likely become the standard for the Web, along with its integrated security features.
Java stands out in three areas of enterprise distributed computing:\(^{60}\)

- **client/server application development** - it is a new, simpler way to develop internal client/server applications.

- **data access** - Java is an easy way to deliver business information broadly, potentially both internally and to external customers simultaneously.

- **on-line transactions** - Java can be used to develop true transaction-based applications, either for business-to-business or business-to-consumer interactions.

However, companies that write virus detection software are predicting that the threat to Java will be worms, programs that replicate in the memory of one machine and are transported over the Internet to replicate on another machine, rather than viruses due to Java’s security design.\(^{61}\) Due to a concern in Java’s security, potential users may wait several months or, possibly, even a year before seriously implementing Java at their site. Java security flaws are being discovered. Users are being urged to only access trusted sites.

Sun Microsystems Inc. has recently announced (February 1996) that they plan to develop a family of microprocessors exclusively for Java applications.

Java has received extensive media attention. Vendors that have purchased licenses for and are incorporating Java applets into their products include: Silicon Graphics Inc., Symantec Corp., Borland International Inc. (Latte), Metrowerks Inc., Netscape Communications Corp.(Netscape Navigator), Oracle (WebServer), Lotus Development Corp. (WebExplorer for Lotus Notes), Powersoft Corp., IBM (beta testing Java on OS/2, AIX, MVS and Microsoft Corp.’s Windows 3.x operating systems), Adobe Systems Inc. (Acrobat electronic document), Novell Inc. (NetWare network operating system and world wide web server), Spyglass and Microsoft Corp.

Even so, there are still a few Java skeptics. They are saying we need to apply a common-sense approach to the current Java craze. They state that if you listen to the Java hype, you hear that Java will enable a new class of computers, revolutionize application development, and be the future of client/server computing; however, there are few relevant Java program examples found on the Web. The examples that can be found are nice and entertaining, but hardly serious applications. They say that Java is just another programming language and not a miracle cure.\(^{62}\)


\(^{61}\) “Boza clowns with Windows 95; experts look for Java worms”, Talila Baron and Norvin Leach, PC Week, p. 14, February 12, 1996.

\(^{62}\) “Beware of Java That’s Too Hot to Handle”, Mark L. Van Name, Bill Catchings, PC Week, p. N22, January 29, 1996.
3) Currently, VRML worlds cannot be produced through a text editor; instead, the user must suffer through CAD or walkthrough programs to construct a world. These tools vary in complexity. VRML also does not scale particularly well -- the real world is not page-oriented, rather, it consists of an inexhaustible and innumerable supply of objects. On the other hand, 3-D worlds have proven very effective for training purposes (i.e., assembly line workers.)

Some VRML browsers cannot retrieve documents on their own -- they have to send their request to another Web browser, which in turn, sends the request on to the intended recipient. The response is returned through the same path.


Java, in conjunction with VRML, appears to be the way to go. People are using Java to add animation and behavior to static VRML worlds. As VRML virtual scenes are downloaded, the associated Java applets would also be downloaded. The applets would then start to execute and give the room life. Without Java, VRML will remain static and lifeless.

Java is something that needs to be learned -- how it works, its security mechanisms and distribution methods, and user access modes. The biggest challenge will be keeping pace with the changes and expansion in capabilities and technologies. For example, at the May JavaOne Developers Conference in San Francisco, Sun unveiled JavaOS, an operating system, and a set of component programming interfaces known as Java Beans.63

A web site designer needs to keep in mind that not all Net users are there (on the Net or accessing their site) for entertainment -- they do not have time for that. They (the users) go to get information and get it quickly and efficiently. Snazzy graphics and animation are inappropriate in some instances.

This new technology is also going to force the developers to become familiar with HTML, Java and VRML (along with associated authoring tools) to keep Net surfers (those out browsing strictly for entertainment purposes) coming back.

63 “Java’s Reach Expanded”, Clinton Wilder with Caryn Gillooly, Information Week, p. 28, June 3, 1996.
6.0 REFERENCES


“Boza clowns with Windows 95; experts look for Java worms”, Talila Baron and Norvin Leach, PC Week, p. 14, February 12, 1996.


CIAC Notes, March 18, 1996, Number 96-01


“Introducing HTML 3.2”, http://www.w3.org/pub/WWW/MarkUp/Wilbur/features.html


“Java Applets”, http://www.javasoft.com/applets

“Java and Internet Programming”, Arthur van Hoff, Dr. Dobb’s Journal, August 1995.


“Java to integrate with enterprise databases”, The Sun Observer, p. 4, April 1996.


“Java’s Reach Expanded,” Clinton Wilder with Caryn Gillooly, Information Week, p. 28, June 3, 1996.


“Navigator Scripting”, http://home.netscape.com/eng/mozilla/Gold/handbook/javascript/script.html#C1


“The One Hour Java Applet”, Philip David Meese, Datamation, p. 51 - 69, March 1, 1996.


“Puzzle me this, puzzle me that”, Rebecca Sykes, February 15, 1996, JavaWorld Online, http://www.javaworld.com/cgi-bin/w3com/start?JW+main


“Sun, Netscape introduce JavaScript object scripting language”, The Sun Observer, p. 6, February 1996.

“Sun Opens Door to Java Development”, Richard Karpinski, Communications Week, p. 1, 70, January 8, 1996.


“What is This Thing called Java?”, Authur van Hoff, Sami Shaio, Orca Starbuck, Datamation, p. 45, 46, March 1, 1996.


“3-D technology hits the Web”, Richard Karpinski, Interactive Age, October 1995.
7.0 TERMINOLOGY/GLOSSARY

anchor - a hotlink from one HTML document/VRML scene to another.

applet - Java - a Java program that can be transmitted over a network and run on any client.

betweening - VRML - moving model between key frames.

big endian - encoding scheme - larger order bits are stored in the lower ordered memory spaces.

bytecode - Java - implementations for the class methods written in the instruction set of a virtual machine; machine independent code generated by the Java compiler and executed by the Java interpreter.

class libraries - Java - collections of object-type definitions from which one can create objects in programs (see packages).

constructor - a method that creates an object.

culling - VRML - removing polygons from the renderer that are not visible to it.

depth - VRML - amount object appears to come towards or recede away from the user.

flat shading - VRML - entire polygon face is shaded with same value so that 3D characteristics are not apparent.

height - VRML - top-to-bottom attribute of an object.

instancing - VRML - using the same instance of a node multiple times.

key framing - VRML - poses; specify position of each model and path in scene.

node - VRML - a VRML object.

normal - VRML - a line that passes through the surface of a polygon at a right angle.

packages - Java - prewritten classes for use in building programs.

parse tree - VRML - the parser “walks” the data within the VRML file and converts the file into this internal representation.

pitch - VRML - object orientation change due to looping/slanting.
point cloud - VRML - the collection of points used to create an object.

point lights - VRML - light sources that radiate in all directions equally from a single source.

polygon - VRML - type of object that makes up a surface.

position - VRML - place where the object is in respect to x- y- and z-plane coordinates.

rendered - VRML - a visible representation is presented on the computer’s display.

roll - VRML - object orientation change due to rolling/side-to-side movement.

scene graph - VRML - list of nodes.

shading - VRML - amount of light with from a source appearing to shine on an object.

surface - VRML - makes the object appear solid; shape that defines an area.

texture - VRML - applying surface attributes to an object so that it does not appear flat.

virtual machine - Java - takes Java compiled bytecode and runs it as if it were machine language.

width - VRML - side-to-side attribute of an object.

wireframe - VRML - linking of points together creating a framework for an object.

yaw - VRML - object orientation change due to spinning about the vertical axis.
The following table summarizes the capabilities/limitations of HTML, Java and VRML in some selected areas.

<table>
<thead>
<tr>
<th>Topic Area</th>
<th>HTML</th>
<th>Java</th>
<th>VRML</th>
</tr>
</thead>
<tbody>
<tr>
<td>inheritance (classes, objects)</td>
<td>none</td>
<td>extensive</td>
<td>limited with respect to object behaviors and cause-and-effect relationships</td>
</tr>
<tr>
<td>security</td>
<td>none</td>
<td>extensive -- very structured (4 passes)</td>
<td>none</td>
</tr>
<tr>
<td>browser support/capabilities</td>
<td>need upgrades to support version enhancements</td>
<td>not all support -- support is growing</td>
<td>not all support -- use of helper applications -- support is growing</td>
</tr>
<tr>
<td>development/authoring tools available</td>
<td>extensive</td>
<td>base growing rapidly -- infancy stage (crude and immature)</td>
<td>base growing rapidly -- infancy stage (crude and immature)</td>
</tr>
<tr>
<td>commercial product support</td>
<td>extensive</td>
<td>base growing rapidly</td>
<td>base growing rapidly</td>
</tr>
<tr>
<td>navigation</td>
<td>hot links/hypertext</td>
<td>none necessary</td>
<td>dependent on browser and hardware (i.e., mouse, glove, etc.)</td>
</tr>
<tr>
<td>main goal/objective</td>
<td>text/publishing</td>
<td>animation</td>
<td>3-D graphics</td>
</tr>
<tr>
<td>animation</td>
<td>no, but HTML documents can contain animation with the &lt;APP&gt; tag</td>
<td>yes</td>
<td>keyframing and betweening</td>
</tr>
<tr>
<td>code generators/convertors</td>
<td>extensive</td>
<td>base growing - infancy stage</td>
<td>base growing - infancy stage</td>
</tr>
<tr>
<td>tutorials/guides</td>
<td>extensive</td>
<td>extensive</td>
<td>extensive</td>
</tr>
</tbody>
</table>

49