An Investigative Study and Comparative Evaluation of Contemporary Three-Dimensional Modelling Using the Virtual Reality Modelling Language

A study submitted in partial fulfilment of the requirements for the degree of Master of Science in Information Management

At The University of Sheffield

By Jin Jie Jun

September 2002
Abstract

This dissertation describes the development of VRML, its characteristics, and its current application in various sectors, and demonstrates the importance of VRML in the contemporary Web-connected community. It introduces the concept of VRML modelling tools, which facilitate the creation of 3D worlds, and explains the great diversity in this sector and the resultant need for a systematic survey and evaluation of the tools currently available. Based on a study of the characteristics of VRML world creation, and the adaptation of an international standard model for evaluating software quality, a set of criteria for evaluating VRML creation tools is established. These criteria are designed as the basis of a detailed evaluation of a VRML tool’s features, an assessment of its advantages and disadvantages in various capacities, and of the skills it may require of a user. A systematic selection of a sample of VRML tools for comparative evaluation is then made, and the tools evaluated according to the criteria established. The results are presented for reference in the form of a comparative matrix, and are then considered in the context of some of the core issues concerning the VRML modelling technology currently on general release.
Acknowledgements

I would send my greatest and deepest thanks to Benjamin Charlton, without his help, this dissertation would be impossible to finish. Many warm-hearted course mates and friends have given me great selfless support in my most difficult time during the dissertation period although they themselves were very busy working as well. There are so many names that it would take nearly a whole page to list them, so I would rather send my deepest thanks to every individual one of them here: Thank you all so much, without your support I would probably have given up!

I would also like to thank my dissertation supervisor Miguel Nunes for his supervision and invaluable advice.

I would like to thank Mr. Garry Beene from WinPlace, Andy Colebourne from AC3D, and Joerg Scheurich (aka MUFTI) from White Dune who responded promptly to my email enquiries and provided invaluable technical information necessary for my system requirements analysis.

Finally I would like to thank my family, my father, my mother, and Emily Lin. It is their selfless love and encouragement that give me the strength and power to finish this hard work.

Thank you all everyone!
Contents

Chapter 1 - Introduction ......................................................... 1
  1.1 Introduction ........................................................................ 1
  1.2 Research Question ............................................................. 2
  1.3 Research Aims and Objectives ............................................. 3
  1.4 Research Methodology ........................................................ 4
    1.4.1 Literature Review: Critical vs. Explorative ...................... 4
    1.4.2 Data Collection: Qualitative Approach vs. Quantitative Approach ...... 5
    1.4.3 Establishing Criteria ...................................................... 5
  1.5 Methodology to be adopted ................................................ 7
  1.6 Reporting ............................................................................ 7
  1.7 The Structure of this Dissertation ...................................... 8

Chapter 2 – Literature Review .............................................. 10
  2.1 Context: The Internet and the World Wide Web .................. 10
    2.1.1 The Internet ................................................................. 10
    2.1.2 The History of the Internet ......................................... 11
    2.1.3 The World Wide Web ................................................. 11
    2.1.4 HTML ...................................................................... 12
    2.1.5 VRML ........................................................................ 13
  2.2 Virtual Reality Modelling Language ................................... 13
    2.2.1 What is VRML? .......................................................... 13
    2.2.2 The History and Development of VRML ....................... 15
    2.2.3 The Future of VRML .................................................. 21
  2.3 VRML Browsers .............................................................. 22
  2.4 VRML Tools ..................................................................... 24
  2.5 Applications of VRML ...................................................... 24
    2.5.1 Education ................................................................. 25
    2.5.2 Medicine ................................................................. 26
    2.5.3 Chemistry ............................................................... 27
    2.5.4 Manufacturing ......................................................... 30
  2.6 Summary ........................................................................... 32

Chapter 3 - Selecting VRML Tools ....................................... 33
  3.1 What Are VRML Modelling Tools? .................................... 33
  3.2 Searching Sources ............................................................ 33
  3.3 Testing Platform ............................................................... 35
  3.4 Initial Selection ............................................................... 35
  3.5 Secondary Selection ......................................................... 40

Chapter 4 - Formulation of Evaluation Criteria ..................... 47
  4.1 Characteristics of 3D World Creation Using VRML ............... 47
  4.2 Definition of Functions to Be Tested ................................... 49
    4.2.1 Creation of Simple Objects ......................................... 50
    4.2.2 Application of Appearance to Objects ............................. 50
    4.2.3 Transformation of Objects .......................................... 51
    4.2.4 Creation of Complex Objects ....................................... 52
    4.2.5 Animation ............................................................... 52
## Tables

Table 1 - Tools Identified in the Initial Selection ..................................................37
Table 2 - Tools Excluded by Secondary Selection ..................................................42
Table 3 - Tools Identified for Evaluation ..............................................................46
Table 4 - Characteristics of Software Quality (ESSI-SCOPE, 1997a) ..................60
Table 5 - Example Comparative Matrix ..............................................................67
Table 6 - Evaluative Summary Table ..................................................................104
Table 7 - Comparative Matrix ..............................................................................106
Table 8 - Comparative Matrix ..............................................................................114

## Figures

Figure 1 – A sample of VRML 1.0 code .................................................................17
Figure 2 – A sample of code from VRML 2.0 .........................................................20
Figure 3 - Event / Route Model (Schneider & Martin-Michiellot, 1998) ..........54
Figure 4 - The Six Characteristics of Software Quality (ESSI-SCOPE, 1997a) ...60
Figure 5 - Criteria Categories Diagram ...............................................................64
Figure 6 - The Interface of AC3D ........................................................................77
Figure 7 - The Interface of ISB ............................................................................78
Figure 8 - The Interface of RenderSoft .................................................................79
Figure 9 - The Interface of Spazz3D ...................................................................81
Figure 10 - The Interface of White Dune ...............................................................82
Figure 11 - The Interface of WinPlace .................................................................84
Chapter 1 - Introduction

1.1 Introduction

HTML and the World Wide Web have revolutionised the way many people obtain information. However, the information contained and displayed in HTML web pages is limited to either two dimensions or static two-dimensional representations of three-dimensional phenomena. This led to the development of what many believe to be the next stage of the current “information revolution”, the Virtual Reality Modelling Language (VRML), “a file format for describing interactive 3D objects and worlds” (Web3D Consortium, 1997).

VRML is a multi-platform language used to publish 3D Web pages. As a simple text language for describing 3-D shapes and interactive environments, it is sometimes regarded to be a three-dimensional equivalent of HTML. VRML is recognised as an international standard by the International Organisation of Standardisation and the International Electro-technical Commission (Web3D Consortium, 1997).

VRML creates three-dimensional virtual worlds and objects, which provide the user with a rich experience of receiving information from the computer. Because VRML displays information in a fashion that simulates dynamically the three dimensions in which human beings naturally operate, a user is able to interact with information contained in VRML files in an intuitive way. As a technology that is cheaply and easily accessible, VRML was in its early days thought to have the potential to become the new standard form in which information is exchanged over the World Wide Web.

However, for various reasons this potential has failed to manifest itself. In other spheres however, VRML has recently enjoyed a resurgence and the potential of its application is beginning to be tapped. VRML has assumed a role as one of the major file formats for storing simulated 3D information. Particularly in the educational sector, VRML has many advantages as an economical and accessible simulated 3D technology. Chemical, biochemical and anatomical simulations are areas where enthusiasm is particularly strong. In the manufacturing sector some companies have
begun to employ VRML as a standard format for mechanical part modelling, providing an intuitive and easy to access database of virtual models.

1.2 Research Question

Although VRML is a relatively new technology, there already exist many software applications to facilitate the creation of VRML files. These applications (hereafter referred to as “tools”) were each designed to fulfil different requirements, and they display a huge variety of different functions, features, capabilities, and different requirements in terms of the technical knowledge needed to operate them. A professional attempting to operate in the area of VRML may thus face difficulties in deciding which tool is best suited to his or her purposes, and what type of skills will be required in order to operate the tool effectively.

These problems, while not unique to the field of VRML, are particularly acute in this area due to the way in which this field developed. The development of VRML and its associated applications has not always been a particularly professional or disciplined process. VRML is a concept that has captured the imagination of many independent individual contributors, and its development has often continued independently of commercial or even usability concerns. Therefore, there is a need to survey and evaluate what is available in a systematic manner and present this information in a clear and accessible form.

This dissertation aims to investigate the current status of VRML technology and its applications in order to conduct a comparative evaluation of some of the major VRML authoring tools. The results of this comparative evaluation will provide the reader with a detailed survey of the tools’ features, an assessment of their advantages and disadvantages in various capacities, and an appraisal of the skills they require on the part of the user. This will give the reader an idea of what tools are available at present, and furthermore allow an assessment of each tool’s ability to meet particular needs.
The Information Studies Department of Sheffield University has long shown interest in the potential of the emerging technology of VRML to provide students with a virtual learning environment. It may provide students with a richer learning experience and allow lecturers to make more efficient use of their time. These purposes and interests are the motivation for this choice of dissertation topic. This dissertation will provide: an investigation of the nature of VRML, supported by related knowledge; an evaluation of a selection of typical, widely available VRML creation tools to allow assessment of their suitability to the needs of the Information Studies Department.

1.3 Research Aims and Objectives

The aims of this dissertation are to:

- Investigate the tools and packages currently available for 3D data manipulation and modelling on the base standard of VRML;
- Carry out a comparative evaluation of these tools and packages

Specific objectives for this study will be:

- To identify a list of companies and organisations that are involved in 3D modelling;
- To identify a list of available tools and packages for VRML modelling, and for each of these…
  - To assess the solutions put forward for the creation of VRML worlds;
  - To assess the user skills required to operate the VRML tools;
  - To assess the costs and maintenance issues of those models.
To assess the implications the above hold for the suitability of each product in various capacities.

1.4 Research Methodology

A research methodology is a set of principles of method that are relative and unique to the specific study (Checkland, 1981). The principles of the methodology will guide the whole research study throughout. It is therefore essential to establish these principles clearly from the outset.

Selection of the most appropriate approaches from various alternatives must be determined by the aims of this project. On the basis of the approaches thus determined, general strategies will be produced. These in turn will form a systematic foundation for more detailed methods and strategies that will then be developed.

1.4.1 Literature Review: Critical vs. Explorative

A literature review is the necessary first step as it provides the basis for a thorough understanding of the problem, and will influence the shape of subsequent stages of the methodology.

The function of an explorative literature review is to locate, catalogue, integrate and summarise the existing literature relevant to a particular problem, clarifying the key issues. As opposed to a critical literature review, its function is primarily to gather information in order to deepen understanding of the subject (Taylor & Procter, 2001). However, it may contain a critical element in terms of the fact that gaps in existing knowledge (either generally or within individual items) may be located as a result, and that current conflicts within the relevant sphere will also be identified.
1.4.2 Data Collection: Qualitative Approach vs. Quantitative Approach

Of the two approaches to data collection, the quantitative approach may appear at first glance to be most appropriate in a comparative study, allowing direct comparisons to be drawn. However, while the creation of an index can allow numerical data to be generated even for such subjective factors as "usability", the purpose of this investigation is not simply to rank the software and judge which is "best" according to narrow pre-defined criteria. Rather, it is to investigate in detail the relative merits and demerits they demonstrate in various capacities. Thus, the qualitative approach is considered the more appropriate.

1.4.3 Establishing Criteria

The criteria of comparative evaluation will be established on the basis of the literature review.

i) Generative Questions

Generative questions are used in qualitative social research methods. In this research the different tools being investigated are so different that it may be difficult to draw a general picture of what is to be compared. Rather than arbitrarily selecting evaluation criteria immediately, key questions are first asked in order to then generate in more detail the criteria for a thorough comparative evaluation. Baumgartner & Payr (1996) used such a generative questions approach to evaluate educational software. They claimed that applying this method permitted useful insights into the effective uses of the software, because generative questions "open up the problem space, draw attention to the problematic points and make solutions comparable". (Baumgartner & Payr, 1996, page36) Because of the effectiveness of this approach in this educational software evaluation, this research will adopt a similar approach towards the comparative evaluation of VRML tools. The criteria of comparison will then be formulated as a comparative matrix on the basis of these generative questions.
ii) Comparative Matrix

In this research, generative questions will be used to open up issues pertaining to the research question. However, a general descriptive comparison for each tool may not give the reader a comprehensive and systematic picture of the advantages and disadvantages of each tool and a direct comparison will also be difficult. Therefore, a comparative checklist matrix will be employed to display the results in an accessible form.

Miles (1994) identifies the checklist matrix as a method of qualitative social research. The comparative checklist matrix that will be used in this research consists of a column for each of the tools to be compared, and rows containing the features to be compared (as raised by the generative questions). This allows a direct and intuitive means of comparing the different tools with regard to particular features, as well as a general overview of any particular tool.

iii) Selection of Software

After the investigation of currently available tools a method of sampling for use in qualitative social research will be adopted to select suitable tools for comparative evaluation. Of the various types of sampling method, ‘purposive sampling’ will be adopted – criteria will be established and all the products identified in the search will be screened through these criteria in order to determine the final selection. Elements of so-called ‘convenience sampling’ are also present as a part of these criteria, as the non-existent research budget limits the potential sample to those products which can be tested free of charge. (Bailey, 1994)

iv) Comparative Evaluation

The selected products will first be individually evaluated according to the criteria defined in the comparative matrix. In order to accomplish this, the documentation (where it is provided) will be surveyed. However, there are severe limitations of document study in this specific research area (Bailey 1994). Even if documents do exist, they will lack a standard form, which may make comparison difficult, and in
any case they are unlikely to provide all the information required for a thorough evaluation. Moreover, when dealing with material about a product that is published by its manufacturer, no matter how technical its content may appear to be, there is always some risk of bias. Therefore, the products themselves will be tested on a test platform.

1.5 Methodology to be adopted

The research of this dissertation will be a qualitative survey, and the following detailed stages will be worked through as the methodology to be used:

1. A broad search of the topic area (VRML) as well as the literatures related to the topic.

2. The identification of available 3D data manipulation and modelling tools and packages. Filter the identified tools and packages, and use defined criteria to select several of the most popular solutions to examine.

3. The establishment of judgement criteria to examine and compare these tools’ effectiveness and efficiency.

4. The collection of relevant data.

5. An analysis and assessment of the collected data.

1.6 Reporting

The results and findings of this research will be presented in the form of this dissertation.
1.7 The Structure of this Dissertation

In “Chapter 1 - Introduction” the concept of VRML has been introduced; the need for research in this area explained; and the specific aims and objectives of this research have been outlined. The research methodology and the rationale for adopting these approaches have been given.

In “Chapter 2 - Literature Review” the literature pertaining to this area will be surveyed and a summary of the content given, in order to build up the necessary background knowledge. Based on the information gained from the literature search, the following information, as necessary background knowledge in this research, will be presented: The context in which VRML came into being, as an integral part of the ongoing development of the Internet and the World Wide Web, will be summarised; An introduction to the different versions VRML will be given, including some technical information; A brief explanation of VRML browsers and creation tools will then follow; The chapter will then conclude with some examples of the practical applications of VRML in the areas of education, medicine, chemistry and manufacturing.

“Chapter 3 - Selecting VRML Tools” will explain the criteria and process by which tools were selected for evaluation in this research, and present the final list of selected tools.

“Chapter 4 - Formulation of Evaluation Criteria” will explain the formulation of the criteria used to evaluate the selected tools, based upon a study of the capabilities of VRML worlds and an international standard model of software quality. Based upon these, a series of generative questions will be asked, from which the contents of a comparative matrix will be derived.

“Chapter 5 – Evaluation of VRML Tools” will contain the evaluation of the tools according to the criteria established in the previous chapter. The results will be displayed both as a written evaluation and in tabular form (the comparative matrix).
“Chapter 6 – Conclusions” will discuss the significance and implications of the results, as well as an evaluation of the methodology used in this investigation and suggestions for future research.
A search for relevant literature was carried out both over the Internet and in the university library. Resources available on the Internet were search engines such as Google (http://www.google.com), Web of Science (through Sheffield University Library portal http://www.shef.ac.uk/library), Sheffield University Library online databases and electronic journals, the Web3D Consortium (http://www.web3d.org) and many others. Since specific books on VRML are relatively few, resources in the library consisted mainly of journal articles. A VRML newsgroup also exists, at the address: dp-news.Maxwell.syr.edu / comp.lang.vrml. This newsgroup was useful in building up an understanding of VRML, particularly as a source of expert consultation regarding technical questions.

2.1 Context: The Internet and the World Wide Web

2.1.1 The Internet

The Internet is a decentralized network of many smaller computer networks linked together. It operates on a common system of rules for exchanging information, the Transmission Control Protocol/Internet Protocol (TCP/IP) (Buchanan, 1997).

In its current form, the Internet provides access to an unsurpassed array of information, including Web documents, software, images, audio files, electronic mail, newsgroups and various Internet “chat” facilities. The ability to both acquire and disseminate information that is offered by the Internet serves the many and varied purposes of almost countless individuals and organisations. Governmental, personal, commercial, educational, political, professional, recreational and religious agendas are all operating. The Internet also provides a fast and economical way to communicate over vast distances, and is now accessible to some extent in most areas of the world.
2.1.2 The History of the Internet

The Internet was conceived in 1964 to serve the American military during the height of the Cold War (McKeown & Watson, 1997). It offered a means of ensuring that communication between military units in different parts of the country could continue in the event of a nuclear war. Because it was a decentralized network, even if key locations were targeted and destroyed, communications could continue throughout the surviving parts of the network.

During the 1970s, civilian researchers also began to exploit the Internet as an inexpensive means of communicating between different institutions. Up until 1991, the Internet was subsidised by the United States’ National Science Foundation, which restricted its use to non-profit, educational and governmental institutions. In 1991 these restrictions were lifted, allowing the Internet to be exploited for commercial use, and triggering its explosive growth into the vast self-sustaining global network it has now become. (Handley & Crowcroft, 1995)

2.1.3 The World Wide Web

Although it is a relatively recent development in the history of the Internet, by far the most popular application on the Internet is the World Wide Web. Today’s World Wide Web constitutes, in practice, a vast repository of information, and allows individuals and organisations to locate and obtain information they require with a speed and economy that is has never been known before.

The original Web was developed in 1989 at the European Laboratory for Particle Physics (CERN) as a means of allowing physicists to communicate with colleagues about their work while it was in progress rather than having to wait for completed projects to be published in the usual journals (Maurer, 1996). It was a collection of inter-connected documents that allowed a reader to jump between documents at will via “hypertext”, “a method of linking related information in which there is no hierarchy or menu system.” (McKeown & Watson, 1997, page 10)
The World Wide Web consists of documents called “Web pages”. The distinguishing feature of Web pages is that they are interlinked by a system of “hypertext links” (often abbreviated to “hyperlinks” or “links”) (McKeown & Watson, 1997). As in the original Web, these are commands in the form of sections of text that allow instant cross-referencing with other documents on the Web. Depending on the format of a Web page, hyperlinks may be displayed in the form of menus or lists of “links”, or alternatively may appear as highlighted sections within the main text of a document. They may even appear in the form of a visual representation, such as a picture, that hides the underlying text altogether.

Web pages are stored on computers called “servers” which are connected to the Internet by their owners in order to make these documents accessible to other Internet users. When a person wishes to view a Web page, their computer (referred to as the “client”) will send a request to the server, which will then send back the document requested. The document will be sent as a text file that contains both the text of the document and its format in the form of coded “tags” (which resemble nonsensical words and punctuation). The content of this file is then interpreted by a program called a Web browser on the client computer, which displays it in the form of the original document, complete with layout and formatting. (McKeown & Watson, 1997)

2.1.4 HTML

The Hypertext Mark-up Language (HTML) is the language in which all Web pages are written. It is written as plain text, which means that Web pages can be created using any standard text file editor. The basic component of HTML is a “tag”, which specifies how the browser will display data. Web pages as displayed by a browser can also contain external “multimedia” items (such as pictures or diagrams) whose content is not part of the Web page HTML code, but whose reference details are specified therein, allowing it to be sent and displayed along with the Web page as part of the viewable document. HTML, then, has three chief functions: it describes how a Web browser should display the file it receives from the server, it describes
hyperlinks, and it defines multimedia objects included in the Web document. (McKeown & Watson, 1997)

2.1.5 VRML

HTML and the World Wide Web have changed the way many people obtain information. However, the information contained and displayed in HTML web pages is limited to either two dimensions or static two-dimensional representations of three-dimensional phenomena. In this sense, the Internet and modern computer technology has a potential that the World Wide Web in its current form does not exploit. There are many fields in which information can be communicated much more effectively if three dimensions are used. Particularly in the areas of science, engineering and architecture, the potential offered by a three-dimensional equivalent of the World Wide Web is enormous. Such reasoning led to the development of what many heralded to be the next great stage of the current “information revolution”, the Virtual Reality Modelling Language (VRML), and this is the subject of the following section.

2.2 Virtual Reality Modelling Language

2.2.1 What is VRML?

VRML (pronounced as either ‘V-R-M-L’ or ‘vermal’) is an acronym for the Virtual Reality Modelling Language. As defined by the Web3D Consortium (1997), the body in control of its development, VRML is “a file format for describing interactive 3D objects and worlds.”

VRML can be regarded as a 3D analogue to HTML, and is a multi-platform language used to publish 3D Web pages. It was designed for use over the Internet, Intranets, and local client systems. However it is not a page specification language such as HTML, nor a general programming language such as C or C++, nor a script language.
such as JavaScript. VRML is a technology that integrates two dimensions, text, three dimensions and multimedia into one coherent model. According to Nadeau, D. R. (1998), it is a simple text language for describing 3D shapes and interactive environments. The Web3D Consortium (2000) claims that a 3D metaphor presents a natural user experience that supports classic 2D desktop models as well as extending into broader contexts of space and place.

According to the Web3D Consortium (2000), VRML, in spite of its meaning Virtual Reality Modelling Language is technically neither virtual reality nor a modelling language. This is because virtual reality would imply 3D input devices (such as digital gloves) and 3D experience (such as a head-mounted display). VRML does not require such input devices, nor does it provide such a rich 3D experience. Also, a true modelling language would contain much richer geometric modelling primitives and mechanisms. VRML contains a minimum of geometric modelling features but provides many other features beyond the scope of a modelling language (Web3D Consortium, 2000). Simply, VRML is rather a 3D interchange format, which defines most of the semantics used in today’s 3D applications such as geometry, material properties, texture mapping, animation, light sources, viewpoints and others.


Nadeau, D. R. (1998) and many others have speculated that the 3D world model would supersede the conventional 2D desktop model and that VRML would be a key technology that would shape the future the World Wide Web. Nevertheless, before this replacement becomes a reality, there are still many challenges to be confronted, such as 3D user interface and navigation, user training, and 3D graphics performance.
2.2.2 The History and Development of VRML

i) VRML 1.0

The inception of VRML took place in 1994. Mark Pesce and his colleague developed a prototype three-dimensional interface to the Web, named Labyrinth (Web3D Consortium, 1998). He was invited to present his paper on this prototype at the First International Conference on the World Wide Web. In the conference, the attendees agreed that there was a need for a common language to specify 3D scene descriptions, which could be deemed a three-dimensional equivalent of HTML. The term VRML was first coined then.

VRML was initially the abbreviation for Virtual Reality Markup Language as it corresponded to the HyperText Markup Language (HTML), which is a lingua franca for publishing hypertext on the Web (World Wide Web Consortium, 2002). However, in order to reflect VRML’s three-dimensional modelling ability, the meaning of the letter ‘M’ was later changed from ‘Markup’ to ‘Modelling’.

Following the conference, an email list called “www-vrml” was set up to discuss the development of a specification for the first version of VRML. After some debate, the Open Inventor ASCII File Format from Silicon Graphics, Inc. was selected from among a number of proposed formats as the basis of VRML. It supported complete description of 3D scenes with geometries, lighting, materials, ambient properties, realism effects, 3D user interface widgets, and viewers. During the setting up of VRML 1.0 Specification, it was decided that the problem of defining methods for implementing animations and complex interactive behaviours would be tackled later, with the exception of hyper-linking, which is a vital feature for a network language standard. VRML 1.0 is designed to meet the requirements of platform independence, extensibility and the ability to work well over low-bandwidth connections. VRML 1.0 was introduced in May 1995 and a further clarified version was issued January 1996 (VRML 1.0 Specification, 1995).

Therefore VRML 1.0 describes static worlds and allows the following:
- Virtual 3D worlds created from primitive shapes, such as cubes, cones, spheres and text, or from custom defined shapes.

- Material properties including texture maps to be applied to these shapes.

- Initial viewpoints and the ability of the user to examine or move freely through a scene.

- Objects that are clickable hyperlinks to other VRML worlds or documents.

- Multiple light sources.

- Inline objects, where the geometry is defined in a separate VRML file.

- Objects with different levels of detail.

- Object definitions named and reused.

The following is an example of VRML code quoted from VRML 1.0 Specification (1995). It is a simple worldview of a red cone and a blue sphere, with a directional light.

```#VRML V1.0 ascii
Separator {
  DirectionalLight {
    direction 0 0 -1  # Light shining from viewer into world
  }
  PerspectiveCamera {
    position  -8.6 2.1 5.6
    orientation -0.1352 -0.9831 -0.1233  1.1417
    focalDistance 10.84
  }
  Separator {  # The red sphere
    Material {
      diffuseColor 1 0 0  # Red
    }
    Translation { translation 3 0 1 }
    Sphere { radius 2.3 }
  }
  Separator {  # The blue cube
    Material {
```
**ii) VRML 2.0**

Since the VRML 1.0 specification left certain issues unresolved (such as the implementation of animation and interaction), its use was limited to the description of static worlds only. There was soon demand for VRML’s further development through the addition of new features such as interactivity and behaviour. The VRML Architecture Group (VAG) was formed in 1995 from among the VRML community in order to organise the development of VRML more efficiently and formally. A request for proposals for VRML 2.0 fulfilling the following requirements was issued (Web3D Consortium, 1997).

*Authorability*

Enable the development of application generators and editors, as well as translation tools to convert other 3D file formats into VRML files automatically.

*Completeness*

Provide all information necessary for implementation and address a complete feature set for wide industry acceptance.

*Composability*

Provide the ability to use elements of VRML in combination and thus allow re-usability.

*Extensibility*
Provide the ability to add new elements or object types not explicitly defined in VRML.

Implementability

Capable of implementing on a wide range of systems.

Multi-user potential

Should not preclude the implementation of multi-user environments.

Orthogonality

The elements of VRML should be independent of each other, or any dependencies should be structured and well defined.

Performance

The elements should be designed with the emphasis on interactive performance on a variety of computing platforms.

Scalability

The elements of VRML should be designed for infinitely large compositions and therefore to enable arbitrarily large dynamic 3D worlds.

Standard practice

Only those elements that reflect existing practice, that are necessary to support existing practice, or that are necessary to support proposed standards should be standardized.

Well-structured
An element should have a well-defined interface and a simply stated unconditional purpose. Multipurpose elements and side effects should be avoided.

Several companies and organisations participated in the contribution. Out of received proposals from ActiveVRML, Dynamic Worlds, HoloWeb, Moving Worlds and others, the Moving Worlds from Silicon Graphics was selected as the favourite basis for VRML 2.0 (Marrin, C., 2000). After long months of working with amendments and revisions, VRML 2.0 Specification was finally released in August 1996 at SIGGRAPH '96 (Special Interest Group on Graphical Display).

New features present in VRML 2.0 make it more interactive and more realistic than VRML 1.0, which handles static worlds without interaction or object motion. The most significant of the new features are:

- **Enhanced Static Worlds.**
  Sound, movie textures, fog and backgrounds can now be added to a VRML world. There are new ways to define complex geometries such as terrains and extruded shapes. The VRML 2.0 scene graph structure has also been simplified.

- **Interaction and Animation.**
  VRML 1.0 worlds were static. VRML 2.0 allows objects within the scene to move and respond to both user-initiated and time based events. This is achieved through sensors that detect or generate these events, interpolators that describe what should happen when an event occurs and routes which wire everything together. Complex animations and behaviours can be programmed into a VRML 2.0 world using Netscape's JavaScript or Sun Microsystems' Java languages. Collision detection and navigation style information also help to improve the user interaction experience.
Prototyping New VRML Objects.

New VRML properties or objects can be defined and reused using a prototyping mechanism.

(Ashdown & Forestiero, 1998)

The following is VRML 2.0 code representing the same simple worldview of a red cone and a blue sphere with a directional light as was represented in VRML 1.0 in the previous section. (Web3D Consortium, 1997)

```
#VRML V2.0 utf8
Transform {  
  children [  
    NavigationInfo { headlight FALSE }  
    DirectionalLight {  
      direction 0 0 -1  
    }  
    Transform {  
      translation 3 0 1  
      children [  
        Shape {  
          geometry Sphere { radius 2.3 }  
          appearance Appearance {  
            material Material { diffuseColor 1 0 0 }  
          }  
        }  
      ]  
    }  
    Transform {  
      translation -2.4 .2 1  
      rotation 0 1 1 .9  
      children [  
        Shape {  
          geometry Box {}  
          appearance Appearance {  
            material Material { diffuseColor 0 0 1 }  
          }  
        }  
      ]  
    }  
  ]
}
```

Figure 2 – A sample of code from VRML 2.0

There is a major leap from VRML 2.0 and VRML 1.0, and the syntax difference between VRML 1.0 and VRML 2.0 is obvious. This makes them incompatible with
one another. Nevertheless, there are some conversion tools that can automatically translate from one to the other.

**iii) VRML 97**

Following the setting up of VRML 2.0 Specification, the VRML Consortium was founded from the former VRML Architecture Group, and it began cooperation with the ISO and the IEC. According to Marrin (2000), the VRML community had ultimate technical control over the specification, and the ISO formalised it. In December 1997, VRML 2.0 was formally approved by the ISO and became an international standard as ISO/IEC-14772-1:1997, known as VRML97.

With regard to functionality, there is little difference between VRML 2.0 and VRML97. Some changes were made to the wording and layout of the specification document in order to satisfy the ISO requirements. In this research, the name VRML97 will be used and it is believed that the confusion of VRML 2.0 and VRML97 is eliminated.

Looking back at the development of VRML specification, it was deliberately designed to avoid being seen as an extension of HTML, which may have constrained its future development. HTML is used to describe the content, layout and formatting of a two-dimensional document. VRML, on the other hand is a modelling language. It describes the geometry and positioning of objects within a three dimensional space using 3D coordinates. It also specifies the appearance of these objects and other properties such as lighting and interactions.

### 2.2.3 The Future of VRML

X3D (Extensible 3D) is the name of the next generation of VRML. It is an open standard of 3D on the web and, furthermore, an extensible standard that can easily be supported by content creation tools, proprietary browsers, and other 3D applications, both for importing and exporting. X3D addresses certain limitations of VRML97.
The Web3D Consortium (2002a) claims that although VRML97 is extensible by providing the ExternProto mechanism, there is no real mechanism for creating groups of functionality extensions. X3D's component, level, and profile mechanisms allows for this. And while individual browsers can implement profiles by using protos and externprotos, it is not compulsory for all browser companies to include this. Because it is fully specified, content will be fully compatible.

X3D is extensible, which means it can be used to make a small, efficient 3D animation player, or can be used to support the latest streaming or rendering extensions. It supports multiple encodings and APIs, so it can easily be integrated with Web browsers through XML or with other applications. In addition to close ties with XML, X3D is the technology behind MPEG-4's 3D support. (Web3D Consortium, 2002a)

The Web3D Consortium (formerly the VRML Consortium) is currently working on formalising the specification of X3D and the future of VRML is being decided. At the time of writing, it is possible that X3D will become an international standard in August 2002.

### 2.3 VRML Browsers

VRML files end with the extension ‘*.wrl’ (“world”), and they are often referred to as ‘worlds’. Users need a VRML browser to interpret VRML files and thus view 3D worlds and objects. However with some current technology, it can be achieved by using only Java supported web browsers. According to Kirschenbaum, M. (1996) a VRML browser performs two basic functions: parsing and rendering. ‘Parsing’ a VRML file is to read the file’s content and verify that it conforms to the VRML specification. If the file is not a valid VRML file an error message will be shown. ‘Rendering’ is to translate the code’s description of the scene and objects and display them on the screen.

According to Crispen, B. (1998), there are three different types of VRML browsers:
- Stand-alone application
- Helper application
- Plug-In

Stand-alone applications are often included in another program such as a VRML modeller, or they are compiled into an application, which requires programming skills from users. Helper applications do not need to be compiled. When a general web browser (such as Internet Explorer or Netscape) receives a link to a VRML file it launches the helper application in another window to view the 3D world. Plug-Ins display VRML worlds in the web browser’s own window. Crispen, B. (2000) claims that many people find Plug-Ins preferable to stand-alone and helper applications. As the popularity of VRML increases over the Internet, many versions of web browsers have already pre-installed VRML Plug-Ins (Crispen, B., 1998).

When viewing a VRML world on the computer screen, navigation is the basic action and means of obtaining 3D experience. Various VRML browsers do not necessarily share the same functionalities; nevertheless, most of them have common navigation methods such as ‘walk’, ‘fly’, ‘jump’, and ‘examine’. Due to the nature of 3D navigation, a user could easily get lost and lose their sense of position. Although the viewpoint function could help the user go back to the start position or any position pre-defined by the VRML file, it would be convenient and practical to have a map function that shows the user his position in real time.

Three popular VRML browsers are selected to view the generated VRML worlds in the later part of this dissertation. They are:

- Blaxxun Contact v5.1 from Blaxxun Interactive, Inc.
- Cortona v4.0 from Parallel Graphics
- Cosmo Player v2.1 from Cosmo Software (According to the Web3D Repository (2002), Cosmo Player v2.1 is no longer being updated to take
account of changes to operating systems and browsers, nor is it being supported technically. However, in consideration of its current wide popularity, it has been selected as a testing browser in this dissertation.)

2.4 VRML Tools

It is relatively easy to create documents written in a two-dimensional language such as HTML using any basic text editor. Similarly, one can simply write VRML codes with a text editor and save the files with a “.wrl” extension (to indicate that the file has a format that can be read by a VRML browser). However, this “hand coding” requires an in-depth knowledge of the VRML specification. In comparison to HTML, creating VRML files and worlds is much more complicated due to its three-dimensional nature, and the more complex the worlds and objects to be produced, the more complicated and difficult the coding process will be.

Various VRML modelling software applications have been produced, which enable users to create 3D objects or worlds without having to directly edit VRML code. With the help of these applications, creating a VRML world requires less technical knowledge and the modelling process becomes relatively easy even if scene complexity increases.

2.5 Applications of VRML

At the time of writing, the practical applications of VRML are already beginning to be exploited in several key areas. The purpose of this section is to provide an explanation of the practical significance of VRML and a summary of some of the roles to which its special features recommend it. An explanation of the rationale in a selection of examples of recent and current developments in the field of VRML use now follows.
2.5.1 Education

Its ability to be distributed online with reasonably low system requirements is a major advantage of VRML compared with other 3D technologies. VRML produces smaller files, which are more easily transferred over the web. This is a factor strongly in favour of adopting VRML in the field of education, where financial resources may be particularly limited.

i) Virtual Chemistry Laboratory

Ruiz et al. (2002) at the University of Cordoba have used VRML to create a “virtual laboratory” for students of secondary level chemistry, which they believe avoids many of the problems encountered by users of previous virtual laboratories.

The three-dimensional representation of experiments in the virtual laboratory provides the student with a much more memorable learning experience than simply reading the equivalent content in text form. Although by no means a substitute for a real laboratory, is certainly much more economical in terms of time and resources and can play a valuable supporting role.

The virtual laboratory contains a database of the experiments that it is able to simulate, which can be expanded, potentially without limit. Another advantage is that it can be accessed from any platform, simply using a standard web browser. (Ruiz, I. L. et al., 2002)

ii) Virtual European School

Bouras et al. (2001) advocate the use of multi-user distributed virtual environments for a so-called “e-learning” approach to distance learning; an approach that they believe will become increasingly important in an information-based society. They claim that virtual learning environments combine the best features of real world information navigation with the advantages of online navigation. The particular receptivity of the human memory to places and visual cues is combined with the
ability to perform searches and cross-referencing with speed and convenience. Furthermore, multi-user distributed virtual environments confer the advantages of shared space and real time feedback that counteract the problems of isolation often experienced in distance learning.

Bouras et al. (2001) describe the use of VRML as the 3D technology preferred by the creators of the Virtual European School project, an international virtual school aimed at assisting distance learning of school children and designed for use by teachers, students and publishers of educational material. It provides a space where these different users can interact and resources can be collected and shared. VRML, which allows web-based application (essential for this project), and is an internationally accepted standard (also important, as this is an international project), was selected by Bouras et al. (2001) as the medium of creation in their proposed design because it confers the following additional features - key advantages of VRML - that they deemed essential for their application:

“The most significant are events, eventIns, eventOuts, routes, sensors and scripts. Besides its power in creating 3D virtual environments, VRML is compact in size where a complex 3D scene is described within a few Kb. This feature enables the distribution of a virtual environment among a large amount of users without considerable time delay or network congestion. Furthermore, for the dynamic modification of a virtual environment we propose the use of VRML External Authoring Interface... which defines an interface between a VRML world and an external environment. It provides a set of methods that an external application can use to interact with, and dynamically update a 3D scene in real-time.” (Bouras et al., 2001, page 19)

iii) The World Wide Web Instructional Committee

The World Wide Web Instructional Committee have prototyped several game-like educational packages based on virtual worlds, including a simulation of an enormous living cell, complete with all the processes of life, through which the user may navigate. (Slator et al., 1999)

2.5.2 Medicine

Samothrakis et al. (1997) predicted that the creation of VRML and its adoption as a global standard would have widespread implications for the future of medicine. At
the time of writing, such a profound change has as yet failed to materialise. Nevertheless the medical applications of VRML are significant.

In the area of medical research, VRML can facilitate the examination of data by allowing a precise three-dimensional representation of sets of data that may otherwise be difficult to visualise.

Furthermore, in medicine, there arise situations in which a specialist diagnosis is required in a distant place. The construction of three-dimensional representations of medical images, such as brain scans and magnetic resonance imaging scans can contribute to a fast and accurate diagnosis. Such images can be effectively converted to relatively compact VRML format, allowing their swift communication over large distances. The adoption of VRML as a global standard facilitates this potential for a global exchange of information by ensuring that files can be easily converted into a compact and easily exchangeable form that will, in theory, be recognised anywhere. A further prediction concerning the status of VRML as a global standard was that this would act as a stimulus for software developers. (Samothrakis et al., 1997)

Simulated 3D modeling allows detailed, interactive demonstration of medical procedures, allowing students or doctors learning a new procedure to practice them on a simulation which does not involve any risk whatsoever to a patient. Furthermore in the area of anatomical education, functioning, living systems can be represented without the expense and mess of dissections and endoscopes. Such simulation programs have been written using VRML and are already in operation. A number of them have been collected at http://synaptic.mvc.mcc.ac.uk/simulators.html.

2.5.3 Chemistry

i) Molecular Modelling

The problem of how to represent complex three-dimensional structures in two dimensions has plagued chemistry ever since the discovery of molecular structure. The ability to simulate three dimensions using a computer has been a development of enormous significance in the area of molecular modelling.
A number of advantages of VRML stand out. Casher et al. (1998) cited the rich navigation features offered by VRML, as well as the ability to create hyperlinks from individual regions of the molecule to other Internet resources in a manner analogous to HTML hyperlinks.

According to Casher et al. (1998), another great advantage of VRML is that it can be readily extended to displaying more complex chemicals. These often require different symbolic representations to illustrate the key features. In principle, any computed molecular surface can be represented in VRML.

Constructing VRML models enables detailed investigation of the structure and behaviour of molecules. Exploring different ways of representing molecules in three dimensions may lead to insights that two-dimensional representations and crude, physical modelling tools would be unlikely to allow. (Casher et al., 1998)

ii) “Knowledge Discovery” and “Data Mining”

It has been said that “the important thing in science is not so much to obtain new facts, as to obtain new ways of thinking about them". (Casher et al., 1998) The idea of KDD (“Knowledge Discovery in Databases”, also called "data mining") has received a great deal of interest in recent years as vast quantities of information have begun to accumulate in databases. The objective is to find new ways of searching these databases in order to discover unusual relations and correlations between and within the data.

Casher et al. (1998) applied VRML to the problem of representing 3D information "mined" from chemical databases. Casher claims:

“Visual representations can allow certain attributes in the data to be noticed precisely because they are unusual. A VRML-based model permits us to view different types of data collected together in one scene, and to make links between different levels in the data hierarchy to allow probing of any unusual facets that may emerge in a highly intuitive manner.

“The conversion to VRML allows for the vectors, colour coded according to the type of contact group, to be hyperlinked to a second model, which in turn can be used to highlight the significant interaction, and give other information such as"
the unit cell and a literature reference. This reference in turn can be a link to the original electronic journal article which allows the research to progress from the discovery of a potentially interesting effect to reading the original article about it.”

(Casher et al., 1998)

**iii) Animation**

Adding a temporal dimension to a molecular model can enhance perception and understanding of a subtle chemical phenomenon. Simple animations can be accomplished using other molecular modelling tools, such as the Chime plug-in from MDL Information Systems, but this requires a set of molecular coordinates for every frame of the animation, whereas VRML often requires only the starting and ending coordinates of the molecule. The properties of the various objects within the scene are then interpolated at each time step to create the effect of an animation. Casher believes that such a technique is particularly effective when animations of large molecules are required.

**iv) Virtual Laboratories**

By representing objects of an entirely different scale, VRML has another important application in chemistry. This lies in its ability to represent complex laboratory apparatus in a much more lifelike manner than is possible in two dimensions, making it easier for others to reconstruct.

The online pooling of various VRML representations of laboratory components connected together with the help of a VRML authoring tool to form an on-line "storeroom" of VRML laboratory equipment has also been suggested. This would, additionally, offer students an accessible means of becoming acquainted with modern laboratory techniques that might otherwise be too expensive to actually carry out.

(Casher et al., 1998)
2.5.4 Manufacturing

i) Maritime Industries

Beier (2000) is extremely optimistic about the use of VRML within the maritime industries. According to Beier, VRML is extremely cost effective since the only infrastructure required is networked computers, which exist almost everywhere, and the VRML plug-in necessary to view the files is available to everyone. At present there are limitations imposed by network capabilities (long download times for large VRML files describing complex virtual models) and the speed of the user's local computer (responsible for real-time rendering and interactions). However, he believes that the current development trend towards high capacity networks like Internet2 and more powerful desktop and laptop computers with 3D graphics acceleration will sufficiently remove these limitations within the near future.

Beier (2000) has investigated applications in design and manufacturing; in the simulation of ship production processes; in ship motion simulation; in the virtual prototyping of sailing yachts as a means of facilitating the specification of custom features in negotiation with customers.

ii) Airport Terminal Design

The University of Michigan Virtual Reality Laboratory developed a virtual model for a new terminal at Detroit's Metropolitan Airport, a large scale, 1.2 billion dollar construction project.

The VRML model of the terminal’s exterior and surrounding environment was created and continuously updated as the project progressed. It served in design evaluations and as a support tool for a complex decision making process. As an interactive VRML application, the model allowed for the study of design alternatives. The effect of these modifications could be evaluated immediately. Various versions of the VRML model were maintained by the Virtual Reality Laboratory and provided on a password protected Web site for all the many parties involved in this project, including engineers and architects at the building site.
Coordination of various parties over distances proved to be extremely effective using the Web-based approach. Major design decisions were based on insight gained from the VRML model and supported by the fast communication of the design's progress over the Web. (Beier, 2000)

iii) Virtual Interface to Manufacturing

Ressler et al. (1997) have created a system called Virtual Interface to Manufacturing (VIM) that uses VRML to access manufacturing data. This 3D interface is targeted at shop floor assembly line workers without a particularly high level of technical education, and although this aspect was untested at the time of writing, the creators were confident that manufacturing and process engineers would certainly have no trouble, as the VRML interface is contained within the familiar context of a web browser.

One of their goals was to create an intuitive interface through which a user with no previous training or experience could browse for information about products and processes. The interface was designed to be able to answer questions such as “What workstations require part X?” or "What part is supposed to be on the table in workstation Y?" that could be asked in a visual way.

In their system, a VRML object acts as the interface to a database. Queries are coded as URL requests, and the data associated with each is stored in a relational database. The associations between a product and process as well as the specific operations that must be accomplished at any workstation, and quality control activities are all stored in the database.

In VIM, there are two ways to use the VRML models. One is to query the database (“navigation”), and the second is for a particular type of visualization (“manipulation”). For example, to the ability to render a particular part solid and the rest of the product in outline only may be useful when a user is viewing the database output for a particular part and prefers to view a 3D model highlighting that part. (This is an example of “manipulation”.)
The key feature of VIM is that it is primarily intended to enable the user to query for data through the visualization. The visualization is the interface to data.

Ressler et al. (1997) believe that the combination of the infrastructure provided by the Web and the visualization capabilities of VRML offer a unique opportunity to create intuitive 3D front ends to data.

2.6 Summary

This chapter has summarised the background knowledge necessary for understanding this research, based on the information collected in the course of the literature review. It has described how VRML came into being as an integral part of the ongoing development of the Internet and the World Wide Web. It has followed the development of VRML through several versions and its adoption as an international standard, and assessed its current status. The two major types of VRML software - browsers and creation tools - have also been introduced. The chapter then concluded with some examples of how VRML is already finding practical applications in the world today, citing case studies from the areas of education, medicine, chemistry and manufacturing.

The examples of VRML use described in the previous section clearly illustrate how much the requirements, budgets and technical capabilities of VRML users can vary. This further highlights the need for a clear and systematic comparative evaluation of the range of different VRML tools available. In the following chapters, a selection of tools will be made from all those available, and a comparative evaluation of these will be carried out.
Chapter 3 - Selecting VRML Tools

This chapter introduces the process of selecting a sample of VRML tools that suit the evaluation needs of this research.

3.1 What Are VRML Modelling Tools?

As introduced in Section 2.4, VRML modelling tools are software applications that enable users to create 3D objects or worlds without having to directly edit VRML code. With the help of these applications, even complex scenes can be created while only minimal technical knowledge is required to operate them.

There currently exist both applications designed specifically for creating VRML files, and other more general 3D modelling packages that can save files in various formats, including VRML.

(Note: In this dissertation, VRML modelling tool software applications are referred to interchangeably as “creation tools”, “VRML tools”, “authoring tools”, “modelling tools” or simply just “tools”. Sometimes more general terms such as “software” and “applications” are also used. The choice of term does not imply any technical distinction, unless one is specified.)

3.2 Searching Sources

An extensive preliminary search for available VRML tools was carried out on the Internet, as this is the primary source of these technologies and the only channel through which those identified VRML tools can be obtained for comparison and evaluation. The search was carried out primarily on the Web3D Repository, which is situated on the Web3D Consortium website.
A definition of the term “Web3D” and an explanation of the Web3D Consortium are therefore necessary at this point.

The Web3D Repository (2002) uses “Web3D” to describe any programming or descriptive language that can be used to deliver interactive 3D objects and worlds across the Internet. About.com (2002) defines “Web3D” as the term for 3D Graphics that are interactive and presented on the Web. VRML is an example of Web3D.

The Web3D Consortium is an international organisation. It has directed the development of the VRML 1.0 and 2.0 Specifications, which provide the basis for the development of associated applications. The information about Web3D technologies provided by Web3D Consortium is therefore authoritative. Its introduction says:

“The Web3D Consortium was formed to provide a forum for the creation of open standards for Web3D specifications, and to accelerate the worldwide demand for products based on these standards through the sponsorship of market and user education programs.”

(Web3D Consortium, 2002b)

The Web3D Repository (2002) is maintained by the Web3D Consortium, and provides impartial, comprehensive, community resources for the dissemination of information relating to Web3D. Therefore, the selection of tools in this research has been mainly carried out according to the information provided by Web3D Repository.

Apart from the Web3D Repository, there is another online channel where VRML is discussed, which is the ‘comp.lang.vrml’ news group. This is a special online community that provides a forum for people to discuss a variety of topics related to VRML and the technology behind it. The subjects of discussion can range from relatively simple ones, such as which VRML browser is suitable for which platform, to complex cutting-edge issues such as database connection with VRML. Many computer scientists and interested individuals who are keen on the development and application of VRML have been actively participating in this online communication.

Despite the overall authority of the Web3D Consortium, the Web3D Repository does not provide complete coverage of all VRML tools. Some tools being proposed and
discussed in the ‘comp.lang.vrm’ news group are not covered in the Web3D Repository. In this respect, the Web3D Repository might not be an entirely comprehensive information source in the particular area of VRML tools. Therefore the ‘comp.lang.vrml’ news group is also an additional information source that has its own authority.

3.3 Testing Platform

All selected tools and applications are to be tested on a personal computer platform:

IBM 600E Notebook with Pentium II 400 processor

Microsoft Windows XP Operating System

10 GB hard disk with up to 3 GB free space

192 MB RAM

1024x768 screen resolution with 24-bit colour

3.4 Initial Selection

Web3D creation tools listed in the Web3D Repository cover a wide range of 3D technologies, including VRML, Java3D, 3DX, Alice, Macromedia 3DS and others. In this dissertation, VRML alone is the focus. Therefore, one of the requirements of the initial selection process is to filter out tools that do not support VRML. Since the testing platform is Windows based, any of those not compatible with Windows will also be excluded.

Thus, the criteria of the initial selection are:
i) The tool must support the technology of VRML (though it may or may not additionally support other Web3D technologies).

ii) The tool must be able to operate on Windows. (Other platform support is not vital.)

The Web3D Repository includes a total of 74 Web3D creation tools. After filtering all those tools according to the criteria set above, 28 applications remain that fulfil the requirements. The list of all 28 tools identified is shown in the following table (listed in alphabetical order).
<table>
<thead>
<tr>
<th>Name of the Tool</th>
<th>Technologies supported</th>
<th>Platforms Supported</th>
<th>Creator</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000cities Builder</td>
<td>VRML97</td>
<td>Windows</td>
<td>2000cities</td>
</tr>
<tr>
<td>3D Anywhere</td>
<td>VRML 2.0</td>
<td>Windows 98/98SE/ME/2000/NT</td>
<td>Micronet Co., Ltd</td>
</tr>
<tr>
<td>3D Create</td>
<td>VRML 1.0</td>
<td>Windows 95</td>
<td>Darwin 3D</td>
</tr>
<tr>
<td>AC3D</td>
<td>VRML97</td>
<td>Windows 95/NT, Linux, SGI, Solaris and others</td>
<td>Andrew Colebourne</td>
</tr>
<tr>
<td>Art of Illusion</td>
<td>Java-VRML</td>
<td>Windows, Mac, Linux, Solaris</td>
<td>Peter Eastman</td>
</tr>
<tr>
<td>Avatar Studio</td>
<td>VRML97</td>
<td>PC</td>
<td>Canal+</td>
</tr>
<tr>
<td>Canoma</td>
<td>VRML97</td>
<td>Win32, Mac</td>
<td>Metacreations Inc.</td>
</tr>
<tr>
<td>Carrara Studio 1.1</td>
<td>VRML 1.0, 2.0 export, Shockwave 3D, MetaStream</td>
<td>EOVIA Corp., TGS Company</td>
<td></td>
</tr>
<tr>
<td>CiteMap Builder</td>
<td>VRML97</td>
<td>Win32</td>
<td>Trivista</td>
</tr>
<tr>
<td>Cortona SDK version 2.0</td>
<td>VRML97</td>
<td>Windows 95 and above</td>
<td>Parallel Graphics</td>
</tr>
<tr>
<td>DesignSpace</td>
<td>VRML 1.0 - VRML97</td>
<td>Windows 95/NT</td>
<td>DesignSpace, division of ANSYS</td>
</tr>
<tr>
<td>Application</td>
<td>VRML Format</td>
<td>Operating System(s)</td>
<td>Author/Company</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------</td>
<td>--------------------------------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>Dune v0.13</td>
<td>VRML97</td>
<td>Windows 95/98/2000/NT, Linux, Irix 6.5, Solaris</td>
<td>Stephen White</td>
</tr>
<tr>
<td>Extrusion Editor</td>
<td>VRML97</td>
<td>Windows 95/98/Me/2000/NT</td>
<td>Parallel Graphics</td>
</tr>
<tr>
<td>Internet Character Animator v1.0</td>
<td>VRML97</td>
<td>Windows 95/98/2000/NT</td>
<td>Parallel Graphics</td>
</tr>
<tr>
<td>Internet Model Optimizer</td>
<td>VRML97 - DXF</td>
<td>Windows 95/98/2000/NT</td>
<td>Parallel Graphics</td>
</tr>
<tr>
<td>Internet Scene Assembler</td>
<td>VRML97</td>
<td>Windows 95/98/2000/NT</td>
<td>Parallel Graphics</td>
</tr>
<tr>
<td>Internet Space Builder v3.0</td>
<td>VRML97</td>
<td>Windows 95/98/2000/NT</td>
<td>Parallel Graphics</td>
</tr>
<tr>
<td>Online Generator</td>
<td>VRML97</td>
<td>all</td>
<td>n/a</td>
</tr>
<tr>
<td>Pharus 3D Builder</td>
<td>VRML</td>
<td>Windows, Mac</td>
<td>n/a</td>
</tr>
<tr>
<td>PhotoModeler Lite</td>
<td>DXF, 3DS, and VRML</td>
<td>Windows</td>
<td>n/a</td>
</tr>
<tr>
<td>Real2Virtual Modeller</td>
<td>VRML, OpenGL, IBM HotMedia</td>
<td>Window 9X/Me/2000/NT/XP</td>
<td>Real2Virtual</td>
</tr>
<tr>
<td>RenderSoft VRML Editor</td>
<td>VRML 1.0, VRML97</td>
<td>Windows 95</td>
<td>RenderSoft Software and Web Publishing</td>
</tr>
<tr>
<td>SitePad (and SitePadPro)</td>
<td>VRML 2.0 - Java</td>
<td>Windows 95/NT</td>
<td>Modelworks Software</td>
</tr>
<tr>
<td>Spazz3D</td>
<td>VRML97 - X3D</td>
<td>Windows</td>
<td>Virtock Technologies</td>
</tr>
<tr>
<td>Vera</td>
<td>VRML 1.0 - VRML97</td>
<td>Windows</td>
<td>VRML3d.com</td>
</tr>
<tr>
<td>Editor</td>
<td>VRML Version</td>
<td>Supported OS</td>
<td>Compatibility</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>-----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>VrmlPad 1.2</td>
<td>VRML97</td>
<td>Windows 95/98/2000/NT</td>
<td>n/a</td>
</tr>
<tr>
<td>White Dune</td>
<td>VRML97</td>
<td>Windows, Linux, Solaris</td>
<td>n/a</td>
</tr>
<tr>
<td>X3D-Edit</td>
<td>VRML97 - X3D</td>
<td>Windows 95/98/2000/NT</td>
<td>n/a</td>
</tr>
</tbody>
</table>
3.5 Secondary Selection

Due to time constraints, the tools selected above are many more than that can be evaluated, and the selection must be further narrowed down by another round of filtering.

VRML has been developed from VRML 1.0 to VRML 2.0 and VRML97. Although the time period of its development is short, and VRML is relatively new technology, there is a significant difference between the VRML 1.0 specification and the VRML 2.0 specification (though, as explained in Section 2.2.2, there is little difference between VRML 2.0 and VRML97). Not only have many new features been added in the VRML 2.0 specification, but also the syntaxes of some similar concepts have been changed. Consequently it is not meaningful to include in the comparison those tools that support only VRML 1.0.

Since no fund is provided for this research, tools to be selected should be either freeware, or have a demonstration version or trial version that can be tested free of charge. Tools should also be able to be downloaded from the Internet, as this is the only means of acquiring these tools for evaluation.

In the initial selection, tools were selected which were designed to serve specialized areas of VRML application. There are, for example, VRML tools that specialise in generating complex geometric objects, and tools that can convert file formats between VRML and other 3D formats. The tools and applications to be compared in this research will be those created as general world creators designed to allow users to create general 3D worlds and objects.

Therefore, the criteria of secondary selection are:

i) The tool should at least support VRML97 (VRML 2.0).

ii) The tool should be freely available for download from the Internet, and cost free.
iii) The tool should be one created as a general world creator, rather than a VRML tool with another specific function.

The secondary selection results in 5 tools from the Web3D Repository plus 1 tool identified from the ‘comp.lang.vrml’ online newsgroup being chosen. These tools fulfil both the criteria set for initial and secondary selection. Reasons for not selecting other tools are listed in the following table.
<table>
<thead>
<tr>
<th>Name of the Tool</th>
<th>Reason for not being selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000cities Builder</td>
<td>Specialised in building 3D city scenes, the language used is not English</td>
</tr>
<tr>
<td>3D Anywhere</td>
<td>Designed to publish 3D scenes on universal platform without demanding any specific servers, browsers or Plug-Ins. Therefore it does not export VRML, although import is supported.</td>
</tr>
<tr>
<td>3D Create</td>
<td>Specialised in creating 3D avatar from any image files</td>
</tr>
<tr>
<td>Aref of Illusion</td>
<td>Not available</td>
</tr>
<tr>
<td>Avatar Studio</td>
<td>An avatar creation tool</td>
</tr>
<tr>
<td>Canoma</td>
<td>Specialised in creating photo-realistic 3D models from scanned or digital photographs</td>
</tr>
<tr>
<td>Carrara Studio 1.1</td>
<td>Not available</td>
</tr>
<tr>
<td>CiteMap Builder</td>
<td>Specialised in creating 3D landscapes</td>
</tr>
<tr>
<td>Cortona SDK version 2.0</td>
<td>A software development kit</td>
</tr>
<tr>
<td>DesignSpace</td>
<td>A specialist software for engineering</td>
</tr>
<tr>
<td>Software</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dune v0.13</td>
<td>Simple general world creator, but there is a later version, <strong>White Dune</strong></td>
</tr>
<tr>
<td>Extrusion Editor</td>
<td>A visual plug-in for <strong>VRMLPad</strong></td>
</tr>
<tr>
<td>Internet Character Animator v1.0</td>
<td>Specialised in animating 3D characters for the Internet</td>
</tr>
<tr>
<td>Internet Model Optimiser</td>
<td>A model optimiser</td>
</tr>
<tr>
<td>Internet Scene Assembler</td>
<td>Specialised in creating and assembling 3D scenes from static and animated objects</td>
</tr>
<tr>
<td>Online Generator</td>
<td>A bundled online VRML generator; creation is operated on the Internet</td>
</tr>
<tr>
<td>Pharus 3D Builder</td>
<td>Interface is embedded in VRML scene, not a stand-alone application</td>
</tr>
<tr>
<td>PhotoModeler Lite</td>
<td>Specialised in creating 3D models from photographs</td>
</tr>
<tr>
<td>Real2Virtual Modeller</td>
<td>Specialised in generating photo-realistic 3D models from photos, images and drawings</td>
</tr>
<tr>
<td>SitePad (and SitePad Pro)</td>
<td>A text-based editor for VRML programming</td>
</tr>
<tr>
<td>Vera</td>
<td>Link provided not operational; no access to the tool</td>
</tr>
<tr>
<td>VRMLPad 1.2</td>
<td>A text-based editor for VRML programming</td>
</tr>
<tr>
<td>X3D-Edit</td>
<td>Focused on X3D technology</td>
</tr>
</tbody>
</table>
Finally, a total of 6 tools were selected for comparative evaluation. In a previous evaluation done by Ashdown & Forestiero (1998), which is similar to this research, the VRML modelling tool was defined as:

“A stand-alone application with an advanced graphical user interface, that allows the user to interactively create and view custom 3D shapes and worlds, and which can save and reload the scene data in a VRML file format.”

(Ashdown & Forestiero, 1998)

Due to a similar nature and purpose, their definition of VRML tools can be modified and adopted by this research. Consequently, all the 6 tools that have been selected for comparative evaluation in this research satisfy the standard of being Windows-based stand-alone applications with graphical user interfaces, which provide users with the ability to generate three-dimensional objects and worlds supported by the VRML97 Specification. They are listed below:
<table>
<thead>
<tr>
<th>Name of the tool</th>
<th>Company or Individual</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC3D v3.0</td>
<td>Andrew Colbourne</td>
<td><a href="http://www.ac3d.org/">http://www.ac3d.org/</a></td>
</tr>
<tr>
<td>Internet Space Builder v3.0</td>
<td>Parallel Graphics</td>
<td><a href="http://www.parallelgraphics.com/products/isb/">http://www.parallelgraphics.com/products/isb/</a></td>
</tr>
<tr>
<td>RenderSoft VRML Editor v1.72</td>
<td>RenderSoft Software</td>
<td><a href="http://home2.pacific.net.sg/~jupboo/">http://home2.pacific.net.sg/~jupboo/</a></td>
</tr>
<tr>
<td>Spazz3D v2.4</td>
<td>Virtock Technologies</td>
<td><a href="http://www.spazz3d.com/">http://www.spazz3d.com/</a></td>
</tr>
<tr>
<td>White Dune v0.19</td>
<td>Stephen F. White <em>et al.</em></td>
<td><a href="http://www.csv.ica.uni-stuttgart.de/vrml/dune/">http://www.csv.ica.uni-stuttgart.de/vrml/dune/</a></td>
</tr>
<tr>
<td>WinPlace v9.3</td>
<td>Garry Beene</td>
<td><a href="http://www.vbinformation.com/winplace.htm">http://www.vbinformation.com/winplace.htm</a></td>
</tr>
</tbody>
</table>
Chapter 4 - Formulation of Evaluation Criteria

4.1 Characteristics of 3D World Creation Using VRML

In order to evaluate VRML tools, especially their functionality aspects, it is necessary to understand the capabilities of VRML, the features supported by VRML worlds and objects, and the way in which VRML worlds are created. Based on the information in the following sections, Sections 4.1-3, the detailed criteria and comparative matrix for evaluation can be established.

VRML worlds can range from simple objects to highly complex scenes that are extremely life-like representations of real phenomena. This research is not going to target the very highest end of this range, however, but rather reasonably comprehensive VRML 3D building procedures for intermediate level worlds. This will include animation, interaction and other features for enhanced realism.

As explained in Section 2.2 (in the VRML section of the literature review), a VRML file is a text-based scene description. The “node” is the basic unit of VRML. It is used to describe shapes and their properties in the world as well as other functions. There are a total of 54 defined nodes in VRML97 specification together with a self-definable node named PROTO, which allows user to define new nodes. A VRML browser will process and interpret the file and present a three-dimensional visualisation with which the user can interact.

Generally a VRML file consists of four major functional parts: VRML header, scene graph, prototypes, and event routing (VRML Consortium, 1997).

i) VRML Header

VRML header is “a single line of UTF-8 text identifying the file as a VRML file and identifying the encoding type of the file” (VRML Consortium, 1997). A VRML file starts with a VRML header, which must be in the first line of the file. This ensures
that the browser can recognise it as a VRML file. The VRML header in VRML97 is:

“#VRML v2.0 utf8”.

**ii) Scene Graph**

“The scene graph contains nodes which describe objects and their properties. It contains hierarchically grouped geometry to provide an audio-visual representation of objects, as well as nodes that participate in the event generation and routing mechanism.”

(VRML Consortium, 1997)

**iii) Prototypes**

In the words of the VRML Consortium (1997), prototypes “allow the set of VRML node types to be extended by the user” and “may be defined in terms of other VRML nodes or may be defined using a browser-specific extension mechanism”. Prototypes can either be included in the VRML file or be defined externally.

The ability of world creation software to assist in the definition of prototypes is limited by the nature of this process. Nevertheless, it would be helpful for a tool to provide some mechanisms to assist the user in creating new prototypes.

**iv) Event Routing**

Some of the VRML nodes have been assigned “eventIn fields” and “eventOut fields” in their syntaxes, which give them the ability to generate events in response to environmental changes or user interaction.

“Event routing gives authors a mechanism, separate from the scene graph hierarchy, through which these events can be propagated to effect changes in other nodes. Once generated, events are sent to their routed destinations in time order and processed by the receiving node. This processing can change the state of the node, generate additional events, or change the structure of the scene graph.”

(VRML Consortium, 1997)
4.2 Definition of Functions to Be Tested

For the purpose of this research, an “intermediate level” world will be taken as one in which the following characteristics are present. They are listed in order of increasing complexity, and will be tested in this order.

- Creation of Simple Objects
- Application of Appearance to Objects
- Transformation of Objects
- Creation of Complex Objects
- Animation
- Interaction
- Other features

These characteristics can be considered to be the essential aspects of intermediate level world creation. The construction of even the most basic VRML world involves the first three; that is, creating simple objects, applying appearance to them, and placing them in a specific position in the world. A more sophisticated world will further employ more complex objects. The final three items are features that are not essential components of a world. Rather, they are the most important of the new capabilities appearing in VRML2.0, whose inclusion in a world may therefore justify it being regarded as being of intermediate rather than elementary complexity.

An explanation of these features now follows.
4.2.1 Creation of Simple Objects

All visible objects are defined inside the shape node, which has two fields: appearance and geometry (Fernandes, 2002). The geometry field indicates which shape is to be drawn. There are four predefined, primitive geometric nodes in VRML97 specification: **Box**, **Cone**, **Cylinder**, and **Sphere**. There is also a **Text** node, which enables the user to create 3D shapes for each character of one or more lines of text.

Although the creation of simple objects is relatively easy, pure coding certainly requires VRML programming knowledge. Therefore, the basic usefulness of VRML tools depends to a large extent on the effective support of simple object creation and automatic VRML file production. Later in Section 5.4, the comparative evaluation will investigate how effectively the identified tools support the creation of simple objects.

4.2.2 Application of Appearance to Objects

Appearance of objects and shapes in a world can be specified in terms of colour, textures and so on. In a VRML file, a **Shape** node contains exactly one geometry node in its geometry field (VRML Consortium, 1997), while in its appearance field it contains various appearance nodes used to define colour, textures and other appearance features to be applied to the geometry. The appearance field is optional in a VRML file; a set of default values will be used if nothing is specified to appearance (Fernandes, 2002).

Applying appearance to objects can be a complicated process. In the absence of specific in-depth knowledge of the process, it is often beyond a user’s direct ability. For example, colour is applied to objects by defining a floating-point value between 0.0 and 1.0 for each of the Red, Green, and Blue colours. These values determine the intensity of each colour, and the resultant combination of these three colours at the specified intensities is then applied to the object (Fernandes, 2002). This process is far from intuitive, being very abstract and lacking direct sensory involvement. With
the visual representations provided by tools, the task of appearance application becomes relatively easy.

The evaluation in Section 5.4 will look into the extent to which applying appearance to objects is supported and how it is achieved in terms of user operation of the tools.

4.2.3 Transformation of Objects

Like real-world building instructions, the text of a VRML file must include precise measurements for size and position of the shapes that will be “built” by the browser.

The three types of changes to geometric objects - scale, rotation and translation - can be collectively termed “transformation” (Fernandes, 2002). The Transform node is a grouping node that lets the author define a set of nodes as a single object and then effect its transformation. In translation, for example, the author defines the translation field in the transform node by specifying coordinates that describe a geometric object’s position in terms of the three-dimensional axes of a VRML world. In addition, the transform node allows the author to define a new local coordinate system for the nodes within the group, which in the words of Ames et al (1997), is “a very powerful feature of VRML”.

A three-dimensional world created by VRML is hard to envisage if it is known only in terms of figures representing three-dimensional coordinates. Conversely, it is also extremely difficult and time consuming to create a complex world directly using coordinates alone. The extent to which an authoring tool facilitates the transformation of geometric shapes, and the usefulness of the manner in which it represents this, are therefore basic subjects of evaluation.
4.2.4 Creation of Complex Objects

Objects in the real world are rarely limited to constructions consisting of simple geometries such as boxes and cones, and a VRML world created with only basic primitive shapes will have great difficulty approximating real-life 3D experience. Since the philosophy of VRML is to bring realistic three-dimensional experience to the user, the creation of complex objects is supported in VRML97. Similar to creating simple objects, complex objects to be drawn in a world are defined by specifying geometry to be drawn in the hierarchical structure of the shape node in a VRML file. The five complex geometry nodes are as follows: **PointSet**, **IndexedLineSet**, **IndexdFaceSet**, **ElevationGrid**, and **Extrusion**.

More than any function hitherto described, the creation of complex objects requires a level of geometric knowledge that makes it extremely time-consuming, if not actually impossible, without the visual aid provided by modelling tools.

VRML tools that support the creation of complex objects thus make greatly enhanced three-dimensional richness available to the user. Unlike the basic functions outlined above, not all tools incorporate these more advanced complex modelling features. These features will be included in the evaluation carried out in Section 5.4.

4.2.5 Animation

Ames *et al.* define animation as a ‘change of something as time progresses’ (Ames *et al.*, 1997, p109). Animation in VRML consists of changes to a coordinate system’s position, orientation, scale, or other changes. Any animation requires two elements: a clock to control the playback of the animation and a description of how something changes during the course of the animation (Ames *et al.* 1997, p110). VRML97 provides animation by using nodes called “interpolators” and “scripts” (Frerichs, 2002).
i) Key-Frame Animation

There are six types of interpolators (position, orientation, colour, scalar, normal and coordinate) (Frerichs, 2002). Interpolators are built-in behaviour mechanisms that generate output based on linear interpolation of a number key data points (Frerichs, 2002). Use of these key data points to generate animation objects is called keyframe animation (Frerichs, 2002), and is the key concept in interpolator animation. In, for example, the example of animation of an object’s position: the TimeSensor node works as a clock and provides time input for interpolators. After receiving a time event, the PositionInterpolator generates a new position output event and sends it to the Transform node that controls the object’s behaviour of position (About.com, 2002b). The description of animation needs to provide a position for each new time as the animation’s fractional time (cycle time) progresses (Ames et al, 1997). However, since theoretically there are an infinite number of intervals between the beginning and the end of the fractional time it is impractical to provide a position for each of these intervals in order to generate the object’s animation. Key-frame animation specifies position to a few key fractional times (points), and the position interpolator node uses the key fractional times and values as a rough sketch of the animation and fills in the values between those specified as needed (Ames et al., 1997). The same principle applies to the other interpolators that generate animation.

ii) Scripts and Routing Animation

Scripts are used to create more intricate forms of animation behaviour. Script nodes themselves contain source code or pointers to code for logic operations. Users can feed data into a script, and have the script analyse the input and output an event to change the world (Frerichs, 2002).

VRML animation is an enormously complex task, which can be made accessible to a novice user by way of powerful visual tools. The Section 5.4 will investigate ways in which the identified tools can provide this convenience to the user.
4.2.6 Interaction

Interaction is another important feature that makes VRML stand out among 3D Web technologies. The user can interact with the world through the mechanism of sensor nodes in the scene graph. There are two types of nodes: environmental sensors and pointing-device sensors. Environmental sensors are: the **Collision** node, the **ProximitySensor** node, the **TimeSensor** node, and the **VisibilitySensor** node (VRML Consortium, 1997). They respond to the movement of the user through the world or the passage of time to realise interaction. Pointing-device sensors are: the **Anchor** node, the **CylinderSensor** node, the **PlanSensor** node, the **SphereSensor** node and the **TouchSensor** node. These sensors respond to the user’s interaction with geometric objects in a world.

Although interaction with a VRML world can take many forms, the basic principle is the same (Schneider & Martin-Michiellot, 1998). It employs the mechanism of events/routing provided by VRML. When user interaction is sensed a sensor generates output in terms of an event in a specific form associated with it. The event is then routed into other nodes providing the data to amend objects’ positions, orientation or other features, or trigger an animation.

The following figure represents the events/routing mechanism employed in the interaction, taking the example of a light being switched on by sensing a user’s action.

![Event / Route Model](image)

Figure 3 - Event / Route Model (Schneider & Martin-Michiellot, 1998)
Similarly to animation, while the ability to realise interaction in VRML greatly enhances the performance and interest of a VRML world, the creation of it demands of the writer a great amount of VRML programming knowledge, which makes it impractical for naive users. The VRML creation tools exist to facilitate the creation of VRML worlds and bring the experience and even joy of it to authors who lack extensive knowledge of the VRML language. In the later evaluation, the success of those identified tools in achieving this purpose will be assessed.

4.2.7 Other Features and Functions

Apart from the world creation features identified above, VRML97 provides various other features that can enhance the performance of a world and sometimes increase the world’s realism. Although they are not basic requirements for creating a VRML world, they are desirable “extras”. Whether or not these features are supported thus becomes a criterion for evaluating the comprehensiveness of a VRML creation tool.

i) Level of Detail:

The Level of Detail node is usually abbreviated to LOD. It is used to specify different representations of a graphical object according to its distance from the viewer (Fernandes, 2002). The level of rendering required increases as the complexity of an object increases; there is no benefit in displaying an object in full detail when the object in the scene is far away from the viewer. Through the LOD node, the VRML browser is able to automatically switch between different versions of an object based on its distance from the viewer (Ames et al., 1997). This sort of “streamlining” allows scenes to be rendered more efficiently and improves the performance of the world as a whole.

ii) Light:

In the real world, the behaviour of light plays an important part in the brain’s construction of comprehensible scenes from visual sense data, as well as, of course,
being a prerequisite for vision to take place at all. In virtual worlds, the monitor provides the light necessary for physical vision to take place. However, simulated lighting can add a great deal of depth and perspective to a scene. VRML was created in such a way that illumination within a scene occurs entirely in relation to simulated light sources.

VRML supports three types of light: point lights, directional lights, and spotlights. The user can place lights at specific locations and aim them in specific directions. In the words of Ames et al. (1997), “Creative lighting can add tremendously to the realism of any world.”

Since, in a VRML world, a light source is necessary for the viewer to see objects, there is also a default “headlight” automatically created by the browser. This attaches to the viewer’s current viewpoint (although it can be switched off using the browser’s options or with the NavigationInfo node) (Fernandes, 2002).

iii) Inlining:

Inlining is a technique that enables the user to build each of the pieces of a VRML world in separate files. The Inline node provides the file name of a piece to use in the VRML world and the browser assembles the world itself. The use of the Inline node simplifies world design and allows the user to reuse the same parts in many worlds without having to duplicate them.

The user specifies a URL where the data can be retrieved in the Inline node (Fernandes, 2002). Generally the files specified within Inline nodes are on a client’s local hard disk, but the user can also use a full URL to specify the address of a file on the Web. This enables the user to build very complex worlds without having to create each shape himself. (Ames et al., 1997)

iv) Anchor:

Being a Web technology, VRML not only provides the ability to deliver 3D worlds over the Internet, but also provides the ability to link those worlds together from
within any world through a mechanism called an “anchor”. Anchors are built using the Anchor node. The user can define a set of objects as a link to a URL that indicates the Web address of a destination VRML world. When a viewer clicks on an anchored object, the browser follows the link, fetches the destination world and displays it (Ames et al, 1997). Anchors are often used as one way of achieving user interaction.

v) Billboard:

The Billboard node is a grouping node, which creates a special purpose “billboard” coordinate system (Frerichs, 2002). A billboard coordinate system is automatically rotated in accordance with an axis that is defined by the user, so that all the shapes in the group always turn to face the view as a unit while the viewer moves around the group (Ames et al., 1997). The billboard function can be used in various effects such as signs, help messages, annotations, status displays and so on (Ames et al, 1997).

vi) Switch:

The Switch node is a grouping node. It contains a group of child nodes but uses the group as a list of choices and selects only one child node to be built at a time (Ames et al., 1997). A writer could use the Switch node to group together different versions of a shape so that quick change of the shape could be achieved without much change to the VRML file.

vii) Background:

Although the user can use three-dimensional shapes to build sky and horizon features, it is a waste of the browser’s rendering time since the viewer will never fly close to the sky or walk to the horizon. VRML provides the Background node to efficiently create world backgrounds. By using this node, the user is able to control the sky and ground colours and set up a panorama of background images to create distant mountain ranges or cityscapes. The Background node enables the user to provide an
outer environment for the world without having to create 3-D shapes for sky and horizon features. (Ames et al., 1997)

viii) Fog:

The Fog node in VRML enables the user to add virtual atmosphere effect to a world and therefore enhance its realism (Fermandes, 2002). By specifying two attributes, the colour and the thickness, the author is able to control the appearance of the fog added to the world (Ames et al., 1997).

ix) Multimedia / sound and audio clips:

Adding sounds to a VRML world can greatly increase its realism. VRML provides the Sound node, which can be used to introduce sound into a world (Fernandes, 2002). According to Ames et al. (1997), adding sounds involves two components: the sound source and the sound emitter. The sound source provides a sound signal, and the sound emitter turns the signal into sound that can be heard. The sound source can be either wave files/midi sounds with the use of the AudioClip node, or mpeg files with the use of the MovieTexture node (Fernandes, 2002).

It is not only the visual world supported by VRML that is three-dimensional, the audible environment is too. Apart from specifying the sound source, the user can also specify the place where he wants to locate the sound, plus the direction in which the sound source is pointing. These features enhance the experience of sound in a VRML world, and also increase the complexity of including it.

x) World Info:

The node WorldInfo enables the user to declare the title of the world produced and any additional commentary the user would like to provide. The node has no visual impact on the world, but a viewer can use a browser’s menu in order to view the information. (Ames et al., 1997)
This chapter has introduced the essential characteristics of VRML world creation in order to establish the features that are necessary or desirable in a world creation tool. This provides a basis for the evaluation of tools with regard to functionality. However, a comprehensive evaluation will include not only this, but also an assessment of their quality as software applications. It is with this that the following section is concerned.

4.3 Software Quality Model

This section builds upon the evaluation criteria established in the previous section with regard to the functionality of VRML authoring software. In order to provide a more complete evaluation, an international standard software quality model will be adapted to the purposes of this research, in order to establish the criteria for a comprehensive evaluation of the quality of the tools to be compared.

Although quality is a desirable characteristic for all products in all fields of commerce, the quality of software somehow seems to be a difficult characteristic to evaluate. There are various modern methods for software product evaluation that provide “ways to represent, define, and evaluate software quality in a repeatable and consistent fashion” (ESSI-SCOPE, 1997a). The International Standard of ISO/IEC 9126 (ISO, 2001) provides a system of evaluating software quality. This international standard formally defines software produce quality characteristics with the presence of a software quality model, as shown below:
Figure 4 - The Six Characteristics of Software Quality (ESSI-SCOPE, 1997a)

This software quality model, in the words of ESSI-SCOPE (1997b), “is used to specify the required product quality, both for software development and software evaluation”.

Six quality characteristics are the main categories for quality of software, and each of them contains several sub-characteristics, which define in detail what aspects are considered under the category. Detailed characteristics and sub-characteristics are shown in the following table along with a definition.

Table 4 - Characteristics of Software Quality (ESSI-SCOPE, 1997a)

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Subcharacteristics</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Suitability</td>
<td>Attributes of software that bear on the presence and appropriateness of a set of functions for specified tasks.</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
<td>Attributes of software that bear on the provision of right or agreed results or effects.</td>
</tr>
<tr>
<td>Functionality</td>
<td>Interoperability</td>
<td>Attributes of software that bear on its ability to</td>
</tr>
<tr>
<td>Compliance</td>
<td>Attributes of software that make the software adhere to application related standards or conventions or regulations in laws and similar prescriptions.</td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>Attributes of software that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs or data.</td>
<td></td>
</tr>
<tr>
<td>Maturity</td>
<td>Attributes of software that bear on the frequency of failure by faults in the software.</td>
<td></td>
</tr>
<tr>
<td>Fault tolerance</td>
<td>Attributes of software that bear on its ability to maintain a specified level of performance in case of software faults or of infringement of its specified interface.</td>
<td></td>
</tr>
<tr>
<td>Recoverability</td>
<td>Attributes of software that bear on the capability to re-establish its level of performance and recover the data directly affected in case of a failure and on the time and effort needed for it.</td>
<td></td>
</tr>
<tr>
<td>Understandability</td>
<td>Attributes of software that bear on the users’ effort for recognizing the logical concept and its applicability.</td>
<td></td>
</tr>
<tr>
<td>Learnability</td>
<td>Attributes of software that bear on the users’effort for learning its application.</td>
<td></td>
</tr>
<tr>
<td>Operability</td>
<td>Attributes of software that bear on the users’effort for operation and operation control.</td>
<td></td>
</tr>
<tr>
<td>Time behaviour</td>
<td>Attributes of software that bear on response and processing times and on throughput rates in performances its function.</td>
<td></td>
</tr>
<tr>
<td>Resource behavior</td>
<td>Attributes of software that bear on the amount of resource used and the duration of such use in</td>
<td></td>
</tr>
</tbody>
</table>
performing its function.

<table>
<thead>
<tr>
<th>Quality Characteristic</th>
<th>Sub-characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyzability</td>
<td></td>
<td>Attributes of software that bear on the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified.</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Changeability</td>
<td>Attributes of software that bear on the effort needed for modification, fault removal or for environmental change.</td>
</tr>
<tr>
<td></td>
<td>Stability</td>
<td>Attributes of software that bear on the risk of unexpected effect of modifications.</td>
</tr>
<tr>
<td></td>
<td>Testability</td>
<td>Attributes of software that bear on the effort needed for validating the modified software.</td>
</tr>
<tr>
<td>Portability</td>
<td>Installability</td>
<td>Attributes of software that bear on the effort needed to install the software in a specified environment.</td>
</tr>
<tr>
<td></td>
<td>Conformance</td>
<td>Attributes of software that make the software adhere to standards or conventions relating to portability.</td>
</tr>
<tr>
<td></td>
<td>Replaceability</td>
<td>Attributes of software that bear on opportunity and effort using it in the place of specified other software in the environment of that software.</td>
</tr>
</tbody>
</table>

This quality model with six quality characteristics and sub-characteristics covers a comprehensive range of areas and aspects of software quality; it provides a framework for the evaluation of software. Although it does not incorporate any specific requirements for certain software, this quality model is applicable to general software. With regard to this dissertation’s evaluation of VRML tools, as an
internationally recognised and acknowledged standard such a quality model will guide the establishment of evaluation criteria. However some modification and adjustment will be made in order to suit the needs of this study and achieve the aims and objectives outlined in Section 1.2.

Due to the nature of the ISO/IEC 9126 Standard, the quality model provides guidelines not only for the evaluation of software but also for the development of software. Therefore not all the characteristics and sub-characteristics will be referred to in the actual evaluation procedure. The criteria that are set up here will focus primarily on the following aspects of a software quality model: functionality, reliability, and usability.

### 4.4 Generative Questions

Based on Sections 4.1 and 4.2 above, generative questions will be categorised into five sections: system requirements, compatibility, functionality, user interface, and user skill requirements. Their interrelationship follows this logical order, from the core requirements in terms of computer hardware, which are usually invisible, to the actual interface as directly experienced by the user, and finally the skills required of the user himself. This relationship them is depicted in the diagram beneath.
Through these criteria, generative questions can be asked concerning each of them, in order to further open up the situation and form the basis of the evaluation criteria and the comparative matrix.

Generative questions belonging to each sector are:

**System Requirements**

- What are the hardware configuration requirements of the tool?
- What are the operating system requirements?
- Is any additional software needed to use the tool?

**Compatibility**

- What, if any, is the proprietary file format of the tool?
- Apart from VRML97, what other Web3D technologies does the tool support?
- What file formats can the tool import and export?
• How compatible is the tool with different platforms?

• Is the file exported from one tool compatible with another? (This is defined as the concept of inter-authoring compatibility in this dissertation. Different tools tend to have different specialities in terms of VRML world creation, which makes the ability to share and re-modifying files using different tools important.)

**Functionality**

• For what purpose is the tool intended?

• How many VRML features does the tool provide?

• Is the creation of primitive shapes (such as boxes, spheres and cones) easily supported?

• Is there an object library of pre-defined shapes?

• How is the application of appearance to objects supported?

• Is there a material library for use in the application of appearance to objects?

• Is a texture library included?

• How is the transformation of objects achieved?

• Is the creation of complex shapes supported? Can the user define and edit individual polygons or surfaces? Is Boolean operation supported?

• Does the tool support animation? In what way and to what extent does it support animation?

• Does the tool support interaction? In what way and to what extent does it support interaction?
• What other VRML features and functions are supported in the tool?

**User Interface**

• Is the user interface well designed and friendly? Is it easy for the user to find their way around the various functions?

• Does the tool provide multi-angle view windows for easy viewing of the world being created?

• Is there a node tree window (scene graph) clearly showing the user the VRML structure of created worlds?

• Is the window layout customisable?

• Does the tool provide context-sensitive help?

• How easy is it to create objects and scenes?

**User Skills Requirements**

• For what sort of user is the tool intended? For what sort of the purposes will they be using VRML?

• Does the user need to have VRML coding knowledge? If yes, to what extent?

• What skills are required for the user to be able to use the tool?

• Is there any technical support available for the user, such as a company website help or online forum?

**Miscellaneous (additional considerations)**

• What is the cost of the tool?
• How can the tool be obtained?

4.5 Comparative Matrix

Below is the comparative matrix, which consists of row headings consisting of evaluation criteria, and column headings consisting of the names of the software under evaluation. Where these intersect, the result of evaluation of that tool according to that criterion will be written (usually in the form of “yes”/”no” indicating whether or not a particular feature is present) along with brief comments, explanations or other additional information where necessary.

Table 5 - Example Comparative Matrix

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Tool A</th>
<th>Tool B</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Requirements</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware Requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Free Disk Space</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sound Card</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Software Requirements</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating System Compatibility</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solaris</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>File Export</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VRML97</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proprietary File Format</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VRML1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gzipped File</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>File Import</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VRML97</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VRML1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gzipped VRML File</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3D Studio (*.3ds)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### AutoCAD (*.dxf)

<table>
<thead>
<tr>
<th>User Interface</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undo and Redo</td>
<td><strong>Creation of Simple Objects</strong></td>
</tr>
<tr>
<td>Copy and Paste</td>
<td>Simple Objects Editor</td>
</tr>
<tr>
<td>Node Tree Window (Scene Graph)</td>
<td>Text</td>
</tr>
<tr>
<td>VRML Source Code Window</td>
<td><strong>Application of Appearance to Objects</strong></td>
</tr>
<tr>
<td>Multiple View Window</td>
<td>Material Editor</td>
</tr>
<tr>
<td>Customisable Layout</td>
<td>Texture Editor</td>
</tr>
<tr>
<td>Help Content Included</td>
<td>Gallery of Texture</td>
</tr>
<tr>
<td>Context-sensitive Help</td>
<td><strong>Transformation of Objects</strong></td>
</tr>
<tr>
<td>Real-time Rendering</td>
<td>Scale Editor</td>
</tr>
<tr>
<td>Embedded VRML Browser</td>
<td>Rotation Editor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Creation of Complex Objects</th>
<th><strong>Animation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Polygon Editor</td>
<td>Key-frame Animation</td>
</tr>
<tr>
<td>Elevation Grid</td>
<td>Scripts and Routing Animation</td>
</tr>
<tr>
<td>Extrusion</td>
<td><strong>Interaction</strong></td>
</tr>
<tr>
<td>Boolean Operation</td>
<td>Sensors</td>
</tr>
<tr>
<td>Gallery of Complex Objects</td>
<td>Routes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other Features</th>
<th>Level of Detail (LOD)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Light</td>
<td>Inline</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**User Skills Requirements**

VRML coding skill required
Technical Support Available

<table>
<thead>
<tr>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
</tr>
</tbody>
</table>
Chapter 5 – Evaluation of VRML Tools

In this chapter each of the selected tools will be evaluated against the criteria established in Chapter 4. The results of the evaluation will be presented first in written form, and then comprehensively in the form of the comparative matrix, to facilitate quick reference and direct comparison.

It was decided that the written results would be structured according to criteria category rather than tool by tool in order to facilitate a more direct comparison of the tools in various specific capacities. It is hoped that this structure will assist a reader in determining the most appropriate tool if particular characteristics in a specific area (or areas) are required. A separate evaluation of each tool in turn would provide a more concise assessment of each and may assist a reader in building up a more comprehensive picture of each individual tool, but emphasis on the comparative aspect of evaluation may be lost. However, inasmuch as a general overview of each tool may be required, a tool by tool summary of the comparative evaluation (along with possible implications thereof) is provided in the form of an Evaluative Summary in Section 5.6. Furthermore, following the written evaluation, an overview of a particular tool can be easily gained by referring to the comparative matrix.

The results of the evaluation will be given in five sections (Sections 5.1-5), which correspond to the five criteria categories explained in the previous chapter. The evaluated characteristics of all the tools will then be listed together in the final comparative matrix. Note that the order in which these tools are listed is alphabetical, according to their titles, and does not represent ranking of any kind.

The full titles of the six tools selected for evaluation, as described in Chapter 3, are listed below. Henceforth these tools will be referred to using the abbreviations given in bold.

AC3D v3.0 (will be referred to as **AC3D**)  
Internet Space Builder v3.0 (will be referred to as **ISB**)
RenderSoft VRML Editor v1.72 (will be referred to as **RenderSoft**)

Spazz3D v2.4 (will be referred to as **Spazz3D**)

WinPlace v9.3 (will be referred to as **WinPlace**)

White Dune v0.19 (will be referred to as **White Dune**)

The evaluation of the tools is intended to be as comprehensive as possible within the complexity range of intermediate level VRML world creation. However it is unavoidable that some highly advanced features of VRML world creation provided by certain tools may have been omitted. Some of the tools that have been evaluated do include certain highly specialised or advanced features which are not dealt with in this study. This omission is a result of the specific aims and objectives of this dissertation, which were established prior to evaluation, and the fact that the evaluation was conducted systematically within parameters previously defined. The fact that a particular advanced function has not been mentioned here does not necessarily imply that it does not exist.

### 5.1 Comparative Evaluation of System Requirements

Detailed system requirements for each tool are shown in the following tables. The first source from which this information was sought was the tool’s manual. However, not every tool has a manual, and not every manual states the tool’s system requirements clearly. It was decided in such cases that the information required would be sought by emailing technical support correspondents whose details were given in the manual or on the tool’s website. In some cases, the exact requirements were unknown even to the tool’s creator. The ambiguities and idiosyncrasies outlined above have made it difficult to display these results in a standard form, and have also meant incomplete information in some cases. What follows is a list of system requirements which reflects the information available.
There remains one tool, Spazz3D, for which no information about system requirements was available; however, this tool was able to run on the testing platform used (see Chapter 3 for details of the test platform).

AC3D:

1) **Computer:** 486 or better.

2) **OS:** Any with OpenGL (e.g. Windows ‘98 or later), plus most Intel-based Linux

3) **RAM:** (Unknown)

4) **Free disk space:** 2MB

5) **Display:** (Unknown)

6) **Interface:** Mouse and keyboard

7) **Sound card:** N/A

ISB:

1) **Computer:** IBM PC or 100% compatible, 486 or better. Pentium recommended

2) **OS:** MS Win95/98, WinNT4.0

3) **RAM:** 8MB minimum

4) **Free disk space:** 12MB minimum. Recommended size of swap file 40MB

5) **Display:** SVGA/256 colours or higher, 800x600 high colour mode recommended
6) **Interface:** Mouse and keyboard, optional pen

7) **Sound card:** Optional

### RenderSoft:

1) **Computer:** IBM PC or 100% compatible 486 or better, Pentium recommended

2) **OS:** Windows 95/98/NT4.0

3) **RAM:** 16 MB

4) **Free disk space:** 3MB hard disk space

5) **Display:** SVGA/256 colours or better

6) **Interface:** Mouse and keyboard

7) **Sound card:** N/A

### Spazz3D: No information available.

### White Dune:

1) **Computer:** 486 or better

2) **OS:** Windows 95

3) **RAM:** 12MB minimum; 24MB recommended
4) **Free disk space:** 800KB

5) **Display:** 16 Bit minimum

6) **Interface:** Mouse and keyboard

7) **Sound card:** N/A

---

**WinPlace:**

1) **Computer:** IBM PC Processor: 133MHZ minimum

2) **OS:** Win95 with MS Visual Basic6 runtime files installed

3) **RAM:** 16MB minimum

4) **Free disk space:** 10MB

5) **Display:** 640x480 256 colour

6) **Interface:** Mouse and keyboard

7) **Sound card:** N/A

---

From this analysis, it can be seen that none of these tools require high-level configuration hardware, and notably, that a 3D graphics card is not required. This characteristic makes all these VRML tools usable on most personal computer platforms. Given the fact that the typical system configuration of a Windows based PC is, at the time of writing, of a much higher level than the minimum required for these tools, the creation of VRML worlds with help of these tools is accessible to users without expensive, state of the art computer hardware.
However, as the complexity of VRML worlds being created increases, higher hardware configuration, particularly the inclusion of a 3D acceleration graphics card, will greatly enhance performance.

### 5.2 Comparative Evaluation of Compatibility

The chief way in which questions of compatibility relate to VRML tools is in their ability to import and export files of different types. This is what will be evaluated in this section.

**Export Ability:**

All the tools have the essential ability of producing VRML97 files, as this was one of the basic criteria by which tools were selected for evaluation.

While all adhering to the international standard of VRML97, these tools also display much diversity as they vie for leading positions in the industry. Four out of six tools use their own proprietary file formats, and files thus created are unreadable to other VRML tools. The proprietary format of AC3D is '*.ac'; RenderSoft uses '*.rsd'; Win Place uses '*.wpp'; Spazz3D uses '*.spz'; while ISB and White Dune do not have their own proprietary formats.

Although VRML97 became the international standard in 1997, VRML1.0 files are to some extent still used, making the ability to export VRML1.0 files desirable in some circumstances. RenderSoft and AC3D are able to do this.

The ability to export zipped VRML files may also be a useful feature. VRML has the advantage of producing files of relatively small size compared with other technologies, which makes it convenient for distributing 3D worlds over the Internet. However as the complexity of the world created increases, the file size can increase enormously. The zipped file format of VRML files (called gzipp) can reduce the file size by two to three times. This file format is supported by most VRML browsers.
Three of the tools support the export of gzipped VRML files. They are: ISB, RenderSoft and Spazz 3D.

**Import Ability:**

The ability to import various file formats is another feature of many VRML tools. It may not always be the case that the author will use the same tool throughout the process of creating a world; he may wish to exploit the relative strengths of different tools in different areas. There is therefore a need for a tool to be able to import existing files in order to incorporate or modify them. This need has meant that many tools are able to import files not only in standard VRML97 and their own proprietary formats, but also in VRML1.0 format, and even files created using other Web3D technologies.

AC3D and ISB are the most versatile in this regard. AC3D is able to import both VRML1.0 and VRML97 files, and in addition can also import 3D studio files. ISB can import VRML97 and gzipped VRML files as well as 3D Studio and AutoCAD files. Spazz3D is less versatile, but is able to import gzipped VRML files in addition to VRML97. White Dune and WinPlace can import only VRML97 files. The least versatile is RenderSoft, which can import only VRML1.0 files and no other formats.

Information regarding the tools’ compatibility is summarized for reference in the comparative matrix (see Section 5.7).
5.3 Comparative Evaluation of User Interface

AC3D

Figure 6 - The Interface of AC3D

The screen shot above shows the interface of AC3D. It consists of: one control panel, three orthographic views and one 3D view (real-time rendering). View windows can be shown either together or individually. Each view is a “camera” looking at the object from either left, front, top or perspective view. Changes to objects can be made in any one of the orthographic view windows and are reflected in the other views. The user cannot make changes in the 3D view window. The view windows can also be resized. These multiple views can make it much easier for the user to accurately observe and effectively envisage the objects and scenes being created.

There is a hierarchical view window available showing the hierarchical relationship between objects. However AC3D does not provide a node tree window (scene graph) or a VRML source code window, which would show the inter-relationship of the VRML nodes created and the actual VRML file code respectively.
The help function in AC3D is limited. There is no actual help content within the program, but instead a link to an online manual through which a user can find information about the program.

**ISB**

![Figure 7 - The Interface of ISB](image)

The layout of ISB is in some ways more sophisticated and user-friendly. There are various view windows in the basic layout as shown in the figure above.

A plan view window shows either a top view alone or top and side views together. This window is where the user can edit the scene. There is also a calibrated ‘steering wheel’ providing the ability to rotate the plan around its central vertical axis. A perspective view window shows a 3D view of the scene being created.
A scene tree represents the scene graph and the hierarchy of objects in the scene. This helps a user to better understand the structure of the scene and that of the VRML file being created. However there is no VRML source code window.

There are various gallery and toolbox windows, which appear in the basic layout of ISB. These include a shape gallery, object gallery, texture gallery, picture gallery, texture mapping window, and painter window. However, the layout can be customised by dropping or adding these according to the user’s preferences.

The help content of ISB is comprehensive, and includes and index and a search function. There is also a context-sensitive help button on the toolbar, which the user can select and point to any part of the layout, prompting a brief help note to appear. This greatly assists a new user in becoming acquainted with the program.

RenderSoft

Figure 8 - The Interface of RenderSoft
The interface of RenderSoft is relatively simple, as can be seen in the figure above. It consists primarily of a single tool bar and a single view window. The default setting of the view window is a 3D browsing window, although the user can choose different views (top, left side, right side, front, back, and bottom) from the tool bar. The view window (of whichever view is selected) functions as the editing window where the user creates and modifies the scene.

RenderSoft claims as an advantage its ability to allow the user to browse the 3D scene while simultaneously editing it. Nevertheless, the inability to provide multiple views simultaneously can make it difficult for the user to obtain an overall, comprehensive perception of the world he is creating, and may make the creation process both more difficult and time consuming in terms of placing objects.

Furthermore, when an object in the scene is selected, it is not highlighted in the view window, which can make it difficult to determine which object is selected, particularly when there are many objects in the same scene. This problem is compounded by the lack of simultaneous multiple viewpoints.

There is no node tree window or VRML source code window. The only way to view the structure of the VRML file is to save it and use a text editing program to view the actual source code directly.

There is no context-sensitive help to display immediate help notes in pop-up windows, but the help menu is easy to use and provides help topics, product information, and introduces the new features of the current version, v1.72.
In the default layout, there are four simultaneous view windows: top view, front view, right view, and the ‘ISO view’, which is a 3D view of the scene. The user can also select any one or two of these views to be shown at any one time. This greatly aids the user in viewing all the objects and scenes being created and edited, and facilitates perception of the three dimensions involved.

Of all the tools that have been evaluated, Spazz3D has the greatest number of shortcut buttons. Most functions can be operated using these buttons. This extensive use of icons and visual aids may simplify use of the tool. However, when configured in such a way that all tool bars are visible, a screen with resolution lower than 1024x768 will leave little remaining space for the view windows. This means a compromise must be struck between full exploitation of the toolbar system and the necessity of sufficient space to view the scene being created. Nevertheless, the layout can be easily customised through the ‘view’ menu, and toolbars not in use can be hidden.
Spazz3D features a node tree window, which shows the hierarchical structure of the scene being created, indirectly reflecting the structure of the VRML file. However, the actual VRML code source is not shown.

One unfortunate feature of Spazz3D is that although a help menu is present, there is no actual help content; technical support is only available via email, thus necessitating an internet connection. This can prove most inconvenient. For those who are interested in this tool, however, there is a yahoo discussion group dealing specifically with Spazz3D, in which issues relating to its application are discussed.

**White Dune**

![Figure 10 - The Interface of White Dune](image)

The layout of White Dune, shown above, features a scene tree, field view, route view and channel view windows, and tool bars.
The concept of the “route view” window is unique to White Dune. It graphically depicts the concepts of events and routing in VRML. “EventOut” fields in one node have links showing their routings to “eventIn” fields of other nodes. The route view window together with the node tree window give the user a clear idea of what is happening within the structure of the actual VRML file while the scene is being created. This makes White Dune a particularly useful tool for creating worlds that incorporate interaction and animation.

The “field view” window shows relative values indicating the various properties (fields) of an object selected, and is also the place where these values can be changed by dragging the mouse or using the keyboard to type in a specific value.

The node tree window, field view window, and route view window provide insight into the structure of VRML file as it is being created. However, the lack of multiple view windows can make it difficult to gain a good graphic overview of how the scene actually appears.

Nearly all the functions are realised through the use of buttons, and there are no related functions in the menus. For example, when adding a box to the scene, the button in the tool bar is pressed and the box will appear in the 3D view window in its default form, and corresponding nodes will automatically appear in the node tree window at the same time.

The layout is customisable. The size of each window can be adjusted and the user can switch off any of the view windows using the “options” menu.

Undo and redo, copy and paste are supported.

There is no context-sensitive help function, nor indeed any help content within White Dune itself. There is a separate user manual available for download and a White Dune website to which a user seeking further technical support can refer.
Figure 11 - The Interface of WinPlace

The layout of WinPlace consists of two windows: a top view window on the left and a 3D view window on the right. The user creates objects in the top view window (referred to as the “2D drawing pane”), and they are then displayed in the 3D view window, in which the user can rotate and examine the scene being created. Note that there is a default base plane in the 3D view window (as seen in grey in the figure above) and all objects created will appear on this plane. This is something that the user cannot change. Also, the layout cannot be customised besides altering the size of the two view windows.

There is a texture gallery at the bottom of the 3D view window, which contains 18 texture styles for the user to choose from.

There is no node tree window, but there is a VRML source code window, which can be opened from the “VRML” menu. In the VRML source code window, the user can directly observe the VRML file’s code being generated as changes are made to the
scene. This gives the user direct feedback on the scene being created in terms of VRML file code.

The way to execute some functions is not immediately apparent in WinPlace. For example, to add lights or set the viewpoint, the user has to right-click in the 3D view window (which in all other respects is a static representation of the scene, in which commands cannot be executed) and select this command from a pop-up menu. This system is unusual and without any obvious rationale, and may initially make it difficult for the user to locate some functions.

In addition, some particular functions require the use of different buttons in combination. This approach can be inefficient and not at all user-friendly. For example, a function as fundamental as moving objects along the Z axis requires the simultaneous use of three separate buttons: “increase”, “size”, and “offset”. Such awkward methods may be a necessary consequence of an interface that represents three dimensions through a single, static top view alone.

Furthermore, unlike other tools, many of the functions in WinPlace can only be realised through the use of buttons in the tool bar and not through menus. These buttons exist only as icons and do not indicate the names of the commands they represent, thus their meaning may not always be apparent. This fact, coupled with the fact that combinations of buttons are often required to execute commands, and the absence of a menu alternative, make operation of the WinPlace interface almost entirely non-intuitive, and the user may find it impossible to achieve even some of the most basic functions unless explanatory literature has been read.

The application itself contains no help content, but a URL link to the website where the user can find information about the tool, including tutorials. It will probably be necessary to refer to this in advance in order to overcome the initial difficulties that a user can expect to encounter as a consequence of this tool’s distinctive interface.
5.4 Comparative Evaluation of Functionality

**AC3D**

*Creation of simple objects*

AC3D provides the basic functions of producing primitive shapes, which include: box, cylinder, sphere, disk, rectangle, mesh, ellipse, and gridline object. Of these, disk, rectangle, mesh, and ellipse are shapes which are not specified in VRML itself, but which AC3D is able to create in the same way as these predefined simple objects. It also supports the insertion of text into a world.

*Application of appearance*

In terms of editing appearance, there is a palette of 14 predefined colours to choose from, and the user is also able to customise the diffusive colour, emissive colour, shininess and transparency of an object, specifying values by moving horizontal bars. However there is no colour plate giving the user a direct image of the colour being customised, so it requires some knowledge of customising colours by defining floating values between 0 and 1 to mix red, green, and blue. The user can import picture files (*.bmp, *.gif, *.rgb etc.) and attach them to objects as textures.

*Transformation of objects*

The three type of transformation (scale, rotation, and translation) can be achieved either by selecting and dragging an object in the view windows, or defining the relevant values precisely in the “edit” menu.

*Creation of complex objects*

AC3D supports creation of complex objects by means of “polygon”, “polygon line”, and “line” functions. Shapes thus created can then be extruded by click and drag operations using the mouse in one of the view windows. In addition to these functions, AC3D’s speciality is its wide variety of advanced functions to alter the surface of an object. These include “spline”, “spike”, “make hole”, “bevel”, and
“triangulate”. By combining these operations, an enormous variety of complex objects can be created relatively simply.

*Animation*

AC3D does not provide functions to achieve animation.

*Interaction*

AC3D does not provide functions to achieve interaction.

*Other features*

There is no complex objects gallery, but there are three additional modelling features in AC3D, which allow the user to alter an already completed object. They are: fragment, merge and explode. Fragment allows the user to turn a selected object into many objects, resulting in a single object per surface of the original object. Merge combines selected surfaces into one single object. Explode breaks apart the model into a series of surfaces, each a selected distance from its starting point.

Point light can be added to the world, but directional light and spot light are not supported. Characteristics of point light such as colour and brightness cannot be adjusted. The background node is supported, and the user can load image files onto three different planes: front (XY), side (ZY), and plan (XZ).

Copy and paste functions allow the user to avoid having to produce the same objects for more than once. Undo is also supported, but it only extends to the last operation performed, and there is no “redo” option to reverse it.
**ISB**

*Creation of simple objects*

ISB features a “shape gallery” (an inbuilt library of simple object), from which users can select simple objects to create. It contains 13 primitive shapes in the shape gallery, including cube (box), sphere, cylinder, and pyramid. The insertion of text is also supported.

*Application of appearance*

A user can apply colour to objects either by selection from a palette of 24 predefined colours or using RGB (Red, Green and Blue) colour slide bars to specify each of the three colours to be mixed (ISB uses floating values between 0 and 225). A texture library contains 62 patterns which the user can choose to apply to objects.

*Transformation of objects*

The scaling, rotating, and translating of objects are achieved by selecting and adjusting with the mouse. Although relative values of scale, rotation and translation are shown in the status bar, the user is not able to alter these figures directly, and if precise values are required, the object must be manipulated with the mouse until the required value is reached. This absence of a function for direct value definition may have the potential to cause enormous inconvenience in contexts where precision is required.

*Creation of complex objects*

There is a Polygon mode in which the user can draw polygons in the top view window, and then select from a number of options (solid, wall, room, or pyramid) which determine the manner in which the polygon drawn will be extruded into a 3D object. In addition, the user can use a Boolean operation to extract an object from another to make a new shape.
There is an “object gallery” (an inbuilt complex object library), which contains 16 predefined objects, such as a chair, a teapot, a desk and others. This is an easy way to add realism to even simple scenes, and provides an accessible way for a new user to experience the satisfaction of being able to create life-like (if rather generic) worlds very quickly. However, the content of the library in the trial version is extremely limited and may not be particularly useful to an experienced user with more specific or technical aims in mind. Fortunately, however, the version of ISB tested was a trial version; it is claimed that the full, registered version contains over 250 samples in both shape gallery and object gallery. Furthermore, the contents of the complex object library can be supplemented by the importation of files describing complex objects.

Animation

ISB does not provide the functions necessary to achieve animation.

Interaction

ISB does not provide the functions necessary to achieve interaction.

Other features

Lighting the world is not supported in ISB, but among other features ISB supports the addition of anchors to objects, the addition of sound and setting background to the scene. Copy and paste functions are also supported.

ISB features a unique method of editing objects and then adding them to the scene. Since other VRML creation tools have the common method of editing objects once they are already inside the scene, this system may be confusing to user who is already familiar with another tool. Also, the undo and redo function provided only applies after objects have been added to the scene, not during the process of editing objects.

ISB has a special function in terms of view points: a user-controlled “free camera” which can be placed anywhere within the world in order to view objects and the scenes from different angles. In addition a user can specify various view points by
placing cameras in the scene to provide view points from which a viewer will start or can switch between when exploring the world.

**RenderSoft**

*Creation of simple objects*

There are shortcut buttons for simple objects (cone, box, cylinder, sphere, and disk) in the tool bar to allow easy introduction of these objects into the world. The user is also able to insert ASCII text via an ‘edit’ option.

*Application of appearance*

To apply appearance to an object, the user first selects it by clicking it while the ‘pick’ button (on the toolbar) is depressed. The user then presses the ‘colour’ button, and a pop-up window that appears. By sliding three bars representing the RGB colours the user creates a mixture, and then associates the resultant colour with one of three colour type options (diffusive, emissive, or ambient colour). There is no palette of predefined colours from which to choose, and there are no actual floating values indicated on the slide bars; changes to an object’s colour are directly reflected on the object as shown in the main view window.

To edit an object’s texture, the user can load either “*.bmp” or “*.jpg” files onto a selected object by clicking the ‘pict’ (sic) button.

The application of appearance is rather primitive in RenderSoft, which makes it likely that the user will experience difficulties if attempting to specify an object’s appearance very precisely. This is particularly true with regard to the application of colour.
Transformation of objects

To transform an object the user can select it in the view window and use buttons in a toolbar together with mouse movement to move, scale, or rotate it.

As with applying material to objects, none of the three transformations can be precisely specified with numerical values. In addition, as RenderSoft does not provide multiple views of the scene being created, it is often difficult to arrange the spatial relationship of different objects in the scene.

Creation of complex objects

The creation of complex objects is restricted to the insertion and modification of extruded polygons. There is no complex object gallery.

Animation

RenderSoft supports the key-frame animation of objects’ spatial positions. This is achieved by selecting and moving objects to the desired positions in the scene while editing a pop-out animation window. Script animation is not supported.

Interaction

RenderSoft does not support interaction.

Other features

The user can change the colour of the background by specifying RGB colours, but cannot insert image files as backgrounds.

The user can insert point light into the scene and specify its position by clicking and moving it within the view window. Although a light that has been introduced to a scene can be moved at any time, it cannot be deleted once it has been added. The user can select the ‘undo’ function to delete the light, but because only the last operation performed can be undone, this must be done immediately after the light is added.
**Spazz3D**

*Creation of simple objects*

Spazz3D provides four primitive objects: box, cylinder, cone, and sphere. The insertion of text is also supported.

*Application of appearance*

To apply appearance to an object, the user selects the object and uses the ‘edit material’ menu to modify the object’s appearance via a pop-up window. There are 18 predefined colours for the user to choose from, or colours can be customised, either by sampling from a panel containing a full spectrum or by specifying each RGB value numerically.

Another way of applying appearance is to select the object and double-click it. A window showing the properties of the object will appear, in which the user is able to edit the appearance. The object’s texture is also edited in this way, though there is no texture gallery included.

Of all the tools evaluated, such a pop-up window is unique to Spazz3D. It contains the properties of a selected object, including its name, geometry characteristics, transformation (scale, translation, and rotation), material, texture and others. This enables the user to edit an object’s characteristics conveniently, without requiring menu systems and toolbars to be navigated.

*Transformation of objects*

After selecting an object, translation, rotation, and scaling can be achieved by clicking the relevant shortcut button and moving the mouse in any of the view windows. While this is being done, relative transformation values are shown in the status bar on the bottom of the window for the user’s reference. Alternatively, the user can double-
click an object and precisely define the desired values in the pop-up window that appears.

*Creation of complex objects*

Facilities for the creation of complex objects include creating “extrusion”, “swept extrusion”, “sculpted surface”, and “rotation” objects, which is reasonably comprehensive. Boolean operation is also provided by means of extraction, insertion, and union of objects.

*Animation*

Text that has been inserted into a scene can be animated by selecting one of 14 preset actions, such as “tall and short”, “thin and fat”, “twirl back and forth” and others. This is one of Spazz3D’s unique features.

The two methods of animation - key-frame animation and scripts animation - are both supported. Animation is achieved by selecting it in the “create” menu or by clicking a shortcut button. The user can define relative values, scripts and routing options through a pop-up window by double-clicking the “animation” icon in the node tree window. It is relatively easy to achieve animation as all the relevant values and options available are displayed in the same pop-up window.

*Interaction*

The user can place sensors in the scene. There are three types: touch, visibility, and proximity sensors. As with animation, the user can modify relative values and scripts and routing options through a pop-up window by double-clicking on the “sensor” icon in the node tree window.

*Other features*

Undo and Redo functions are supported along with a “history” window listing operations performed. Copy and paste functions are also supported.
To add nodes to a scene, the user can either select them from the “create” menu, or click shortcut buttons on the toolbar. Furthermore, their various relative definitions and values can be edited using as pop-up window, which is one of the unique features of Spazz3D.

Spazz3D covers all of the advanced features of VRML world creation that were identified in “Chapter 4 - Formulation of Evaluation Criteria”. From this point of view, it is the most comprehensive of the tools evaluated.

**White Dune**

*Creation of simple objects*

White Dune supports the following primitive objects: box, sphere, cone, and cylinder. The user clicks the relevant button in the toolbar and the object will be introduced into the scene. Text can also be inserted into the scene.

*Application of appearance*

To apply appearance to an object, there is no menu or buttons in the toolbar as other tools usually use. The user finds a “material” node in the node tree window, and from there is able to open a “field view window” through which he can modify values in various “fields”. These fields define various properties of the object’s appearance, and consist of emissive colour, diffusive colour, specular colour, shininess and transparency.

There is no texture gallery, but the user can load image files to attach to objects as textures. This is done by clicking the texture button in the toolbar.

*Transformation of objects*

When an object created in the 3D view window is selected, axes (in the form of three colour-coded arrows) appear in the centre of the object. The user clicks and drags the
mouse pointer along the arrow that indicates the direction in which he wishes to move
the object. Alternatively, he can define values in the relevant field view window by
dragging the mouse or by entering numbers. Modifications made in the latter way are
represented in real time in the 3D view window, allowing subtle control along with
simultaneous visual feedback.

Rotation and scaling operations follow a similar procedure. The user clicks an object
to select it, and then controls the rotation or scaling by moving the mouse along axes
in the 3D view window.

Creation of complex objects

In White Dune, it is intended that an object’s characteristics be modified primarily by
altering the relevant values in the field view window. Because complex objects
would require the input of an extremely large number of values, the creation of
complex objects is not supported at present. However, White Dune is still under
development and is being upgraded regularly. The toolbar features buttons such as
“IndexFaceSet”, “IndexLineSet”, “PointSet”. In the current version, when these are
clicked, the relevant nodes are added to the node tree, but are not represented in the
3D view window.

Interaction

Interaction could be regarded as a strong point of White Dune. Interaction is mainly
cconcerned with the use of sensors to detect actions the user executes through input
devices such as the mouse. Events generated in response to this are routed into other
nodes in order to trigger actions in the world. White Dune features a route view
window, which clearly demonstrates the relationships between nodes and clearly
shows the processes of interaction in terms of the routing of events. To realise
interaction the user adds sensors, of which there are various types, using buttons in the
toolbar, and then links an eventIn of one node to an eventOut of another node. This
may require a certain degree of familiarity with the concepts of events and routing
used in VRML.
Animation

Key-frame animation is not supported by White Dune. However, more complicated scripts animation can be achieved through modifications executed in the “route view” window, which visually represents the routes through which events are sent. However this requires a high level of VRML coding knowledge.

Other features

White Dune supports many other features, including: Light (all three kinds of lighting are supported), LOD (Level of Detail), Inline, Anchor, Billboard, Switch, Background, Fog, Multimedia, World Info and View Point. Again, these can be introduced into the scene by clicking their buttons in the toolbar and modifying various values in the field view window.

WinPlace

Creation of simple objects

Simple objects that can be created in WinPlace are: cube (box), cylinder, cone, sphere, and text. The user can draw them in a 2D drawing pane and they are displayed in the 3D rendering view window. Numerical values to indicate the size of the object being created are not given, which means that the user cannot accurately determine or precisely specify its dimensions. Precise values can be determined and entered using a VRML source code window, although this is dependent upon the user having the knowledge of VRML coding necessary to do this.

Application of appearance

To apply appearance to objects, the user clicks the “colour” button and selects a pre-defined colour from a palette, or defines custom colours using a spectrum. The user can also apply textures to objects by clicking the “texture” button and choosing one
texture of the 18 available in the texture gallery. However the user cannot load image files as textures to apply to objects.

*Transformation of objects*

The user can scale objects along X and Y axes in the 2D drawing pane (top view window). Scaling along the Z axis is rather more complicated and requires the use of two buttons: “increase” and “Z axis”. With “increase” button depressed, clicking the “Z axis” button lengthens the object along the Z axis. When the “increase” button is not depressed, clicking the “Z axis” button shortens the object along the Z axis.

Rotation of objects can also be achieved in the 2D drawing pane (top view window). However, objects can only be rotated along the Z axis, and not the X or Y axes. This is a limitation of WinPlace.

Translation of objects along X and Y axes is also done in the 2D drawing pane. Similarly to scaling objects along the Z axis, translation of objects along the Z axis requires the operation of two buttons: “increase” and “offset”. With the “increase” button depressed, “offset” moves the object along the Z axis in the positive direction (i.e. away from zero if the current value is positive; towards zero if it is negative), and vice versa.

In all three types of transformation described above, numerical values that indicate their magnitude precisely are not displayed, thus the user cannot specify their values precisely unless manually making changes to the VRML code directly using the source code window.

*Creation of complex objects*

The creation of complex objects is literally not supported in WinPlace. Although the user can edit polygons in the 2D drawing pane, polygons created in this way appear in the 3D view window as primitives, usually boxes, of shape and dimensions apparently unrelated to those of the polygon drawn.
Animation

Animation is not supported.

Interaction

Interaction is not supported.

Other features

Other features include three types of lighting: directional, point, and spot lights. These can be introduced into the scene by right-clicking in the 3D view window and selecting them from the “editor” menu.

The user can also specify a colour for the background, but cannot load image files as background. This is also achieved by right-clicking in the 3D view window and selecting from the “editor” menu.

5.5 Comparative Evaluation of User Skills Requirements

Basic computer literacy is a fundamental requirement for a user of VRML modelling tools. Knowledge of the operation of windows-based applications, including use of the mouse, menu commands and so on, is the basic prerequisite.

The tools investigated in this research were selected on the basis not of being specialised technical applications, but general world creators. As such, they could be regarded as specialising in the area of not requiring any specific user skills in addition to basic computer literacy (the general skills identified above). Generally speaking, they are successful in this. The degree to which they are successful is the subject of this section.

Those few cases where additional technical knowledge is required will be dealt with in this section. Those cases where a more precise, technical alternative is available in
a addition to the basic function will not be cited here (for example, the understanding of RGB values will not be cited as a user skill requirement if the software also supports the selection of colour via a spectrum or palette as an alternative means of colour definition).

Additionally, it should be mentioned in this section that the technical support available for these tools was found to be effective and efficient. The prompt replies that were received in response to email inquiries regarding system requirements are an example which reflects this particularly clearly. In addition, there exist online discussion groups available for people to discuss different aspects of available tools. This means that any minor difficulties in using a tool, which are likely to be a question of familiarity rather than particular user skills, can be resolved relatively quickly and easily.

An assessment of the user skills requirements of each tool in turn will now be made.

**AC3D**

There is no colour plate to provide a direct image of a colour being customised, so familiarity with customising colours by defining floating values between 0 and 1 to mix red, green, and blue will be needed if a user does not wish to be restricted to the palette of predefined colours.

Other than this, effective use of AC3D to create VRML worlds does not require any specific technical knowledge in addition to basic computer literacy. However, it is likely that good knowledge of geometry will be required if AC3D’s extensive facilities for complex object modelling are to be exploited fully.

**ISB**

ISB features a particularly user-friendly and intuitive interface, with comprehensive and powerful help functions. Because of this, a new user can quickly learn how to use the tool effectively, in spite of its apparent complexity. In addition, the extensive
gallery included in the full version facilitates the creation of lifelike worlds without requiring the technical knowledge necessary to create lifelike objects from scratch.

**RenderSoft**

RenderSoft is particularly uncomplicated in its operation and no VRML coding knowledge is required. Help content is included.

**Spazz3D**

Spazz3D claims that the user does not need to know VRML in order use VRML. However, this is not the case with regard to Spazz3D’s interaction and animation functions. Familiarity with VRML scripts and routing concepts and their application is in fact required in order to correctly set up script animation and interaction in this tool. This may be due to the nature of the tasks themselves - which are not something that can always be done automatically by the tool - but it remains the case that sufficient knowledge of this aspect of VRML is required in order to exploit Spazz3D’s full potential.

**White Dune**

White Dune features “route view” and “node tree” windows, which give the user a graphic representation of what is happening within the structure of the actual VRML file being constructed. However, basic familiarity with the structure of VRML may be useful (though it is not an absolute requirement) if this characteristic is to be fully exploited as a useful feature rather than merely ignored as an unnecessary and unwanted complication. The relevant information can be obtained from a user’s manual, which is available for download.

Interaction functions, which require use of the node tree window, may require a certain degree of familiarity with the concepts of events and routing used in VRML. Animation, which is achieved through modifications executed in the route view window, requires a high level of VRML coding knowledge.

**WinPlace**
The user skills required to operate WinPlace are not so much a question of technical knowledge as of operational knowledge specific to this tool alone. Online help and tutorials are available, through which a new user can quickly become familiar with the operation of the tool. Once this has been done, it is relatively straightforward to use.

5.6 Evaluative Summary

AC3D

AC3D provides many different functions for creating objects. The multiple angle view windows greatly assist the process of production and the 3D view window shows the result in real time. It provides more simple objects in its gallery than are defined in VRML itself. In addition, special object modelling functions greatly enhance the complexity of the objects that can be created.

The absence of support for animation and interaction means AC3D does not qualify as a comprehensive intermediate level general world creator according to the standards specified in this research. However, as a specialist tool in the area of static modelling, AC3D contains unique advanced functions and is capable of complex and precise operations. These advanced and extensive object creation functions firmly establish the position of AC3D as a comprehensive object modelling tool.

ISB

ISB’s most notable asset is its user-friendly and intuitive interface. In particular, its comprehensive and powerful help functions make it easy for the user to find their way around and learn quickly how to use the tool effectively. Its most serious disadvantage is its lack of support of animation and interaction functions, which means it is not sufficiently sophisticated to be considered a truly comprehensive package. Nevertheless, ISB is a suitable tool for the creation of static worlds. ISB is
not well-suited to roles that require a high degree of precision or advanced technical functions. However, with its mobile cameras and inbuilt gallery of lifelike objects, ISB greatly simplifies the task of creating lifelike, realistic scenes.

**RenderSoft**

RenderSoft is a relatively simple VRML creation tool. A notable feature is the single view window, which combines the functions of VRML modeller and browser, and allows the user to edit a 3D scene while viewing it as it would appear on the Web. Unfortunately, in practice it can often prove very awkward to create scenes without the aid of multiple view windows. Furthermore, the user cannot precisely define values for an object’s position or appearance, a feature which effectively makes it unsuited to any role requiring precision.

Despite the limitations imposed by the absence of these features, however, RenderSoft is a simple-to-use tool for creating relatively simple VRML worlds. Furthermore, it supports key-frame animation, which is a surprising characteristic to find in what is in other respects a relatively simple tool. This feature may encourage the exploitation of this tool in conjunction with other tools as a means of introducing animation to worlds constructed on otherwise more advanced tools that lack animation facilities. The other principal need RenderSoft is likely to address is that of a VRML novice, as its operation is relatively uncomplicated and no VRML coding knowledge is required.

**Spazz3D**

Spazz3D is perhaps the most comprehensive and sophisticated of all the tools that have been evaluated. It performs a wide range of functions that assist VRML world creation, and has a user-friendly interface. Its one major drawback, though it is by no means alone in this, is the absence of help and content-sensitive help functions, which
can make it difficult for the user to get sufficient support when it is needed. In addition, a certain amount of prior knowledge regarding scripts and routing in VRML is required if the animation and interaction functions are to be exploited to their full potential. Nevertheless, all the functions necessary for intermediate level VRML world creation can be found in Spazz3D.

**White Dune**

White Dune is unlikely to prove entirely satisfactory in the capacity of a stand-alone modelling tool for general world creation. However, the unique concept of the “route view” window is a factor in favour of its use as a rather more specialist tool in the area of animation and interaction. The route view and node tree windows give the user a very clear idea of what is happening within the world they are creating in terms of the structure of the actual VRML file being constructed. However, a degree of familiarity with the structure of VRML is a likely requirement if this characteristic is to be fully exploited as a useful feature rather than merely ignored as an unnecessary and unwanted complication. The necessary knowledge can be obtained from a user’s manual, which is available for download.

**WinPlace**

WinPlace is a crude and simple modelling tool. Few primitives are supported and its inability to create complex objects severely limits this software’s usefulness as a VRML modelling tool.

The interface is essentially simple, but its design is such that the tool is rather complex to use, despite the comparatively simple nature of the functions it performs. The necessity of some form of external tutorial, instruction or technical support is a likely consequence of the apparently haphazard structure of this tool’s interface.
WinPlace supports very basic VRML creation and appears unsuited to the creation of intermediate VRML worlds.

Nevertheless, once the unique construction of the interface has been fully understood, this is a perfectly adequate tool for creating basic VMRL worlds, and would be a reasonable starting point for newcomers to VRML. It is also worth noting that WinPlace is the only one of the six tools to include a window showing the VRML source code of the file as it is created. For most practical purposes, this approach has few advantages (and numerous disadvantages) when compared to more sophisticated equivalents (for example, the node tree and field view windows used by White Dune) which essentially contain the same information but represent it in a more intuitive and visually accessible way. Nevertheless, the presence of a source code window in WinPlace may mean that this tool holds some appeal for those interested in becoming familiar with (or those already familiar with) the VRML language itself.

**Summary table**

The ability to create simple objects, complex objects, and to achieve animation and interaction are the major functions involved in VRML world creation. This table indicates briefly which tools support which functions, along with suggested uses based on the comparative evaluation.

**Table 6 - Evaluative Summary Table**

<table>
<thead>
<tr>
<th>Simple Objects</th>
<th>Complex Objects</th>
<th>Animation</th>
<th>Interaction</th>
<th>Recommended Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC3D</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>ISB</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>RenderSoft</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Spazz3D</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
5.7 Comparative Matrix

The following comparative matrix is designed to present the results of the evaluation in a form that allows direct comparison of the tools according to the established evaluation criteria, providing a comprehensive and systematic picture of the advantages and disadvantages of each tool. The rows contain comparison criteria; the columns represent the tools. This matrix can be used to gain a general overview of any particular tool, or make a comparison of the different tools according to specific criteria.
<table>
<thead>
<tr>
<th>Criteria/Tools</th>
<th>AC3D</th>
<th>ISB</th>
<th>RenderSoft</th>
<th>Spazz3D</th>
<th>White Dune</th>
<th>WinPlace</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Requirements</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware Requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer</td>
<td>486 or better</td>
<td>IBM PC or 100% compatible, 486 or better. Pentium recommended.</td>
<td>IBM PC or 100% compatible 486 or better, Pentium recommended</td>
<td>Not available</td>
<td>486 or better</td>
<td>IBM PC Processor: 133MHZ minimum</td>
</tr>
<tr>
<td>OS</td>
<td>Any OpenGL plus most Intel-based Linux</td>
<td>Windows 95/98/NT4.0</td>
<td>Windows 95/98/NT4.0</td>
<td>Not available</td>
<td>Windows 95</td>
<td>Windows 95</td>
</tr>
<tr>
<td>RAM</td>
<td>Unknown</td>
<td>8MB</td>
<td>16 MB</td>
<td>Not available</td>
<td>12MB minimum; 24MB recommended</td>
<td>16MB minimum</td>
</tr>
<tr>
<td>Free Disk Space</td>
<td>2MB</td>
<td>12MB minimum. Recommended size of swap file 40MB</td>
<td>3MB</td>
<td>Not available</td>
<td>800KB</td>
<td>10MB</td>
</tr>
<tr>
<td>Display</td>
<td>Unknown</td>
<td>VGA/256 colours or higher</td>
<td>SVGA/256 colours or better</td>
<td>Not available</td>
<td>16 Bit minimum</td>
<td>640x480 256 colours</td>
</tr>
<tr>
<td>Interface</td>
<td>Mouse and keyboard</td>
<td>Mouse and keyboard, optional pen</td>
<td>Mouse and keyboard</td>
<td>Not available</td>
<td>Mouse and keyboard</td>
<td>Mouse and keyboard</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------</td>
<td>----------------------------------</td>
<td>--------------------</td>
<td>---------------</td>
<td>--------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Sound Card</td>
<td>N/A</td>
<td>Optional</td>
<td>N/A</td>
<td>Not available</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Software Requirements</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Not available</td>
<td>N/A</td>
<td>MS Visual Basic</td>
</tr>
<tr>
<td>Operating System Compatibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Linux</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Solaris</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Compatability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>File Export</strong></td>
<td>VRML97</td>
<td>Proprietary File Format</td>
<td>VRML1.0</td>
<td>Gzipped File</td>
<td>3D Studio (*.3ds)</td>
<td>AutoCAD (*.dxf)</td>
</tr>
<tr>
<td>----------------</td>
<td>--------</td>
<td>-------------------------</td>
<td>---------</td>
<td>-------------</td>
<td>------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>VRML97</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Proprietary File Format</td>
<td>*.ac</td>
<td>N/A</td>
<td>*.rsd</td>
<td>*.spz</td>
<td>N/A</td>
<td>*.wpp</td>
</tr>
<tr>
<td>VRML1.0</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Gzipped File</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>File Import</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VRML97</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>VRML1.0</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Gzipped VRML File</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>3D Studio (*.3ds)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>AutoCAD (*.dxf)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Feature</td>
<td>Yes (only undo, no redo)</td>
<td>Yes (apply to operations on scenes, not objects)</td>
<td>Yes (undo once, no redo)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------</td>
<td>--------------------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>Undo and Redo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy and Paste</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Node Tree Window (Scene Graph)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>VRML Source Code Window</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Multiple View Window</td>
<td>Yes</td>
<td>No</td>
<td>Yes (not shown simultaneously)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Customisable Layout</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Help Content Included</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Context-sensitive Help</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Real-time Rendering</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Rendering</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>-----</td>
</tr>
<tr>
<td>Embedded VRML Browser</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Functionality**

_Creation of Simple Objects_

<table>
<thead>
<tr>
<th>Simple Objects Editor</th>
<th>Yes (8 simple shapes)</th>
<th>Yes (13 simple shapes)</th>
<th>Yes (5 simple shapes)</th>
<th>Yes (4 simple shapes)</th>
<th>Yes (4 simple shapes)</th>
<th>Yes (4 simple shapes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

_Application of Appearance to Objects_

<table>
<thead>
<tr>
<th>Material Editor</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes (through material node in scene tree window)</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texture Editor</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Gallery of Texture</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>--------------------</td>
<td>----</td>
<td>-----</td>
<td>----</td>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td>Transformation of Objects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scale Editor</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Rotation Editor</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Translation Editor</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Creation of Complex Objects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polygon Editor</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Elevation Grid</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Extrusion</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Boolean Operation</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Gallery of Complex Objects</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Complex Objects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Animation</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Key-frame Animation</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Scripts and Routing Animation</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interaction</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensors</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Routes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other Functions and Features</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Level of Detail (LOD)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Light</td>
<td>Yes (point light)</td>
<td>No</td>
<td>Yes (point light)</td>
<td>Yes (all 3 kinds)</td>
<td>Yes (all 3 kinds)</td>
</tr>
<tr>
<td></td>
<td>Inline</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-----</td>
</tr>
<tr>
<td>Anchor</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Billboard</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Switch</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Background</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Fog</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Multimedia</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>World Info</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Viewpoint</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**User Skills Requirements**

<table>
<thead>
<tr>
<th></th>
<th>VRML coding skill required</th>
<th>No</th>
<th>No</th>
<th>No</th>
<th>Yes (when applying animation and interaction)</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

113
<table>
<thead>
<tr>
<th>Support Available</th>
<th>(online manual &amp; email)</th>
<th>(online manual &amp; email)</th>
<th>(email)</th>
<th>(online forum &amp; email)</th>
<th>(email &amp; manual available for download)</th>
<th>(email)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>$39.95</td>
<td>$78.95</td>
<td>$15.00</td>
<td>$100.00</td>
<td>Free</td>
<td>Free</td>
</tr>
</tbody>
</table>

Table 8 - Comparative Matrix
Chapter 6 - Conclusions

6.1 Summary

This dissertation has investigated the creation of VRML, its characteristics, and its current application in certain sectors, and demonstrated the importance of VRML in the contemporary Web-connected community. It has introduced the concept of VRML authoring tools, which facilitate the creation of 3D worlds, and explained that the great diversity and lack of industry standards in this sector necessitate an investigation and comparative evaluation of those tools.

Therefore, following a study of the characteristics of VRML world creation, and the adaptation of an international model for evaluating software quality, a set of criteria for evaluating VRML creation tools was established. These criteria were designed to provide the reader with a detailed survey of the tools’ features, an assessment of their advantages and disadvantages in various capacities, and of the skills they require of a user. A systematic selection of VRML tools for comparative evaluation was then conducted, and the tools evaluated. The results were displayed in a form intended to allow an assessment of each tool’s ability to meet particular needs, as well as providing an insight into some of the core issues that concern the VRML modelling technology currently enjoying general availability.

6.2 Conclusions

None of the tools that have been evaluated require high configuration hardware to run, nor do they require the installation of any additional software. This means they are all tools which are accessible to the most parties with an interest in this area.

One characteristic that stands out in the results of this evaluation is that although some tools are more sophisticated than others, none are without their own advantages and disadvantages. Some tools have particularly original features, and great strengths in
certain areas, though none are without weaknesses in others, and the areas of strength and weakness are different for each tool. No one tool stands out from the crowd as an obvious leader in the field. No one tool incorporates all the desirable features and clearly stands above the rest as the industry leader.

The representation of three dimensions on an interface that is essentially two-dimensional, in a manner that allows intuitive and effective interaction, is a huge challenge - perhaps the greatest challenge facing the creators of VRML authoring software. It is a problem the solution of which requires considerable originality of thought, and a problem which all the tools that have been evaluated have approached in slightly different ways. This is clearly a field of enormous diversity, and a user’s choice of tool may, in the end, be a question of personal preference as much as how comprehensive a tool it is.

It is not a tool’s interface alone which displays scope for variation. The creation of complex objects is a function for which there is as yet no standardised method. VRML is an extremely flexible standard and often different approaches are employed by different tools. In the course of this research it was found that when different tools were used to create objects that were ostensibly identical in appearance, the structure of the VRML files created was actually quite different.

VRML creation software is an industry is that has not yet become dominated by standard ideas, and people are keen to have new ideas and experiment with new systems. It has not yet stagnated. This may be because it is as yet still a reasonably young technology. However, it has also been possible partly because the VRML standard is a very diverse standard – there are many different ways of achieving the same result - which allows considerable innovation and original thought in its application.

While the current diversity may be encouraging in terms of allowing space for innovation and creative thought, it can make it difficult for a user to determine which tool is the best suited to their needs. Furthermore, diversity can lead to a degree of confusion regarding the nature of different tools’ functions. For example, it was found that those tools that support the creation of complex objects tend to use
different names for what is essentially the same creation method, instead of using the actual node names defined in VRML97.

The near future may see a greater level of industrial co-operation between different companies and individuals, and holds the possibility of the eventual development of an authoring tool that unites and incorporates all the desirable features currently present only in separate tools.

One encouraging feature is the appearance of an innovative ways of representing VRML files, not merely as 3D images, but in forms (such as node trees and White Dune’s route window) that provide the user with an insight into what is being created and how it works, not just in terms of appearance but also in terms of its underlying structure and functions. As the popularity of VRML increases, if it continues to do so, such features may go some way towards preventing the isolation of VRML as a sort of “black box” technology understood only by the initiated and used only indirectly by the majority of casual users.

### 6.3 Evaluation of Methodology

The selection and implementation of methodology has proved successful in this dissertation, and the original aims have been accomplished. The explorative literature review deepened understanding of a subject originally unfamiliar to the author, and made possible the use of an international standard as the basis for the criteria of evaluation. Generative questions categorised in five sections opened up the problem space and guided the creation of a comparative matrix specially tailored for this evaluation. The completed comparative matrix provides a direct and intuitive means of comparing the different tools with regard to key features, as well as comprehensive overview of any one individual tool.
6.4 Future Work

This evaluation was conducted in accordance with criteria established in advance. Nevertheless, given the nature of the subject and the fact that it was conducted by a single individual, it may be that a certain degree of bias resulting from personal preferences is unavoidable. In this evaluation no general users were involved, and therefore the results do not comprehensively reflect the human aspects of, for example, usability, because no independent user experiences were recorded or evaluated. In future work, a user test session to determine the feelings of a sample of independent parties towards the tools concerned would provide more comprehensive results directly reflecting the perspectives of actual users.

Inter-authoring compatibility is defined in this dissertation, but during the evaluation, certain problems were encountered in that the VRML files of the same scene produced by different tools are structurally quite different from each other. This is because of the diversity of VRML97 standard, which provides the possibility of describing 3D worlds of similar appearance using various different ways of coding. Due to time constraints and the complexity of the task, the inter-authoring compatibility of the tools was therefore not systematically evaluated. This is another recommendation for further study in this area.
Bibliography


