VRML: Adding 3D to Network Management

Luca Deri
IBM Zurich Research Laboratory[1]*, University of Berne*[2]

Dimitrios Manikis
IBM Zurich Research Laboratory[3]

The increasing number, complexity, and heterogeneity of network management resources have pushed industry and research to find new ways to visualise information that go beyond classic 2D solutions. Network topologies and layered information models have exposed the limitations of these solutions, demanding more powerful 3D visualisation techniques.
This paper describes a novel approach to the visualisation of network management resources using VRML, a 3D modeling language. VRML has been successfully applied to selected network management problems, showing that network management information can often be presented profitably in 3D format. The ability of modern Web browsers to handle both HTML and VRML allows a simple yet powerful and flexible system for two and three-dimensional network management visualisation to be created.

Keywords: Network Management, 3D Visualisation, VRML, World Wide Web.

1 Introduction

Over the past few years, graphical computer capabilities have been improved significantly. Today, computers come with a high-resolution video board, and graphical user interfaces are the standard way to interact with software applications. Pushed by the game industry and by vertical markets such as computer graphics and medical visualisation, computer manufacturers are producing faster and faster chips and video systems able to draw and animate realistic images and mathematical models. This quick evolution in computer hardware enabled the move from 2D to more realistic 3D representations. After some attempts to create APIs and modeling languages for 3D, Silicon Graphics, a pioneer of computer graphics, released a graphics library called OpenInventor [9], available on many platforms. Its wide acceptance in the industry contributed to the unification of the various existing APIs for 3D visualisation, apart from being used to originally implement VRML, acronym for Virtual Reality Modeling Language [10]. VRML is a modeling language, hence it describes a set of 3D elements usually called a *3D virtual world*. With VRML people can build virtual rooms, towns

[1] IBM Research Division, Zurich Research Laboratory, Säumerstrasse 4, 8803 Rüschlikon, Switzerland. Email: lde@zurich.ibm.com, WWW: http://www.zurich.ibm.com/~lde/.
[2] Universität Bern, Institut für Informatik und angewandte Mathematik, Software Composition Group, Neubrückstrasse 10, CH-3012 Bern, Switzerland. Email: deri@iam.unibe.ch, WWW: http://iamwww.unibe.ch/~deri/.
[3] Current address: National Technical University of Athens, Telecommunications Laboratory, 15773 Zographou, Athens, Greece. Email: jman@telecom.ntua.gr,
WWW: http://www.telecom.ntua.gr/~jman/.

and landscapes specifying how such virtual worlds look. A software application called *VRML viewer* interprets VRML to render the world, allowing people to explore and navigate it. One of the most interesting features of VRML is its ability to link virtual worlds with the Web associating an HTML anchor with each VRML element. This allows users to jump to other VRML worlds and HTML documents and vice-versa, exploring the Web as if wandering through a vast universe.

Besides some rare exceptions, network management visualisation is still limited to 2D. In many cases, classic 2D representations are too limited and do not allow complex information to be represented easily. Recently, the great diffusion of the Web promoted the development of Web-based network management systems, one of which developed by one of the authors, shows how greatly management systems can benefit from their integration into the Web.

The aim of this paper is to demonstrate how 3D visualisation based on VRML can be effectively applied to network management and combined with HTML to build a simple yet powerful and modern management system. This paper covers the design and the implementation of a VRML-based network management visualisation system including its advantages and disadvantages. In addition it demonstrates how selected network management problems can be solved by combining technology developed by the authors with the power of VRML and the Web, whilst avoiding both the need to purchase expensive/proprietary toolkits that run only on specific platforms, and to use specialised tools for each task.

2 The Virtual Reality Modeling Language

A VRML file is a textual description of a VRML world. It describes how to draw shapes, where to place, and how to display them. Each VRML file is composed of: a) VRML header, b) a set of nodes, and c) fields and comments (optional). The VRML header specifies the VRML version. The nodes describe the shapes and their properties in the world being defined. Each node contains the type of the node and a set of optional node attribute fields. Some node types are Cube, Cylinder, and Sphere, whereas attributes are radius, width, and height. VRML files can also contain a camera object which defines the position and the characteristics of the default view of the world. Additionally the LOD (Level of Detail) VRML tag allows VRML viewers to handle big worlds efficiently. LOD specifies which elements have to be displayed and at what level of detail according to the distance of the camera into the current VRML world.

3 VRML and Network Management

This section describes how network management can benefit from 3D visualisation, why VRML has been selected for this purpose and how it has been applied to selected network management problems.

3.1 Why VRML?

The idea to apply 3D visualisation techniques to selected network management problems is derived from the need to represent management information in a way that is as close as possible to reality. Conventional 2D visualisation systems have many limitations, some of them being (c.f. section 3.2):

- a lack of realistic representation of the information whenever such information is implicitly in 3D format since it has to be flattened in order to be represented in 2D;
- a lack of expressiveness whenever a large quantity of sparse information has to be combined in order to build a compound view of it;
- an inability to display topological information as it is in reality.

Besides all this, 3D visualisation offers several interesting advantages. It allows people to represent the information in a way very similar to reality: to change perspective, to move the viewpoint, and to add or eliminate details by getting closer to the information. Beyond these benefits, it is not straightforward to identify how to apply 3D to network management and where to prefer it to 2D. 3D is significantly more computation-costly than 2D and it is usually not platform-independent, in the sense that applications written using standard 3D graphic libraries cannot run unmodified on different platforms. Additionally, the cost to write an application for 3D visualisation is high in terms of development time and expertise. The solution to all these problems has been the adoption of a modeling language because it allows different worlds to be represented easily, leaving to the language viewer the task to visualise the information. VRML has been selected because it is at the moment the standard modeling language which is also well integrated with the Web since a) Web browsers usually come with a VRML viewer, b) VRML files are retrieved using HTTP, the same protocol used by the Web to retrieve documents, and c) it is possible to jump transparently from HTML files to VRML and vice-versa.

The integration of the Web with network management is becoming more and more important since it allows network resources to be managed in a simple, cheap, platform-independent way directly from within a standard Web browser. This seamless integration is also important in the context of a research project called *Webbin' CMIP*. In this project one of the authors has developed a software application named *Liaison* [5], which allows CMIP/SNMP resources to be managed through the Web. In this view VRML has been preferred over more powerful modeling languages like 3DMF [1] because:

- HTML is a simple, platform-independent and elegant way to display information that can be represented in 2D. The ability to jump from HTML to VRML and vice-versa enables developers to use the best visualisation format for each situation;
- VRML can be visualised efficiently on standard PCs without the need to purchase additional custom hardware;
- VRML is a very simple yet powerful language that can be learnt quite easily, hence it enables developers to create new VRML worlds or enhance existing ones without technical knowledge of 3D visualisation because the VRML viewer is responsible for this.

3.2 Applying VRML to Network Management

The use of VRML in the context of network management is derived from the need to represent management information in an effective way, employing either 3D or 2D depending on the situation. The ability of Liaison to handle different file formats makes it easy to use HTML to display 2D information, whereas VRML is used exclusively for 3D visualisation. The following sections cover the some situations where VRML has been applied in order to overcome the limitations of 2D.

Network Topology

Network topology deals with visualisation of network elements. Topology can be either logical or physical. Logical topology is used to visualise how elements are interconnected and what the connections are, without any constraint about element physical location or topological distance. Physical topology instead requires that elements are placed where they are really located and that distance constraints are satisfied. In the case of logical topology, the goal is to display the information in the most readable way. This is in order to show human operators the current network status without adding additional information like element size or distance, which are not meaningful in this context and may confuse operators.
ATM (Asynchronous Transfer Mode) topology is an example where logical topology is used. In ATM the PNNI (Private Network-Network Interface) [8] database can be queried for retrieving the status of connections and additional information such as ATM addresses. This information is useful not only to retrieve the actual network topology but also to determine the load of the active connections, and hence to optimise the global performance by rerouting the overall network traffic.

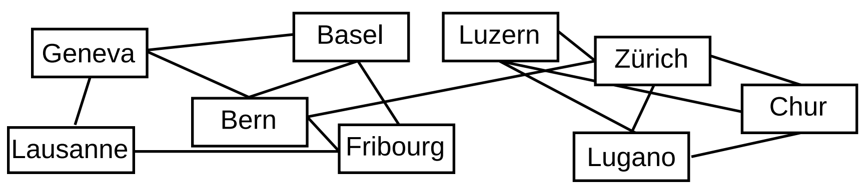

Fig. 1: 2D ATM Logical Topology

The picture above displays a typical 2D ATM topology. While it shows what the operator is supposed to see, it does not highlight important information like the fact that Zurich and Bern are the two main ATM centres in Switzerland. Additionally, in order to display such topology inside a HTML browser, there are two nontrivial solutions: an HTML imagemap or a Java [7] applet. VRML instead allows HTML anchors to be associated with each basic element, hence providing a high degree of control with very low granularity. The following picture shows how the previous ATM logical topology has been implemented using VRML.

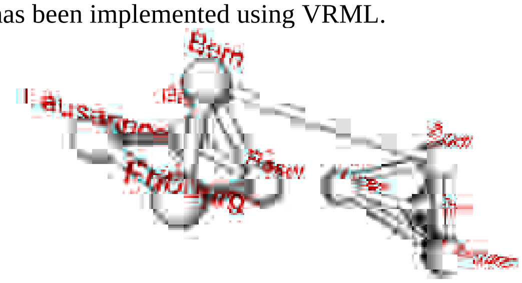

Fig. 2: VRML-based ATM Logical Topology

The Z axis allowed Zurich and Bern to be placed higher than the other ones, hence to highlight the fact that such centres are the main ones in the country contributing to the elimination of line crossings. Each node has a corresponding 3D label and an HTML anchor which enables users to jump directly to an HTML page containing detailed information about it. Connections are depicted using a different colour according to the current link load and it has a corresponding HTML anchor just like the nodes. Users can freely rotate and walk through the world by exploiting the facilities offered by the VRML viewer, hence going beyond vertical/horizontal scrolling available for 2D representations. These advantages become much more evident when a large topology having multiple connections among nodes is to be displayed. LOD allows big worlds to be manipulated efficiently since the VRML viewer does not have to render elements that are too distant from the current camera or represent objects at a high level of detail which is useful when the camera is close to an object. Additionally, LOD allows subnetworks to be represented as one node when the camera is distant, and the architecture of the subnetwork to be visible when the camera is close enough to it.

Hierarchical Information

Quite often in network management, information is aggregated in hierarchical structures. In the OSI world, object instances are stored in a so-called containment tree. This means that an object instance can contain one or more subinstances in a tree structure. In SNMP the same structure can be obtained splitting the information according to the MIB groups, which are identified using nested object identifiers. The TCP/IP protocol identifies hosts assigning addresses of the type X.Y.W.Z, where X, Y, W and Z are integers; host addresses are then grouped in subnets according to a subnet mask. These examples show some simple cases of hierarchical information just to demonstrate that this kind of information structure can be encountered quite often in network management.

While 2D could be used to represent hierarchical data, it has the disadvantage that a) it does not respect the native structure of the information whenever there are connections between the different information levels, making the picture difficult to read, and b) it creates trees that may become very wide or tall if the tree contains many nodes. VRML, instead, can represent the information in its native hierarchical format. This is very convenient whenever the quantity of information to be represented is high, making it possible to create real 3D topologies instead of flat ones. Also, by using the LOD facility, the information about the subinstances in the containment tree can be hidden, shown only when the camera is close to the object.

Another situation to which VRML can be applied is to represent subnetwork connection-termination points. As described in [6], important concepts needed to describe networking are layering and partitioning. They allow a network to be partitioned into a hierarchy of subnetworks with successive levels of abstractions. The following figure shows the partitioning of a network connection into link connections and subnetwork connections (SNC) as for modeling according to the [5] way.

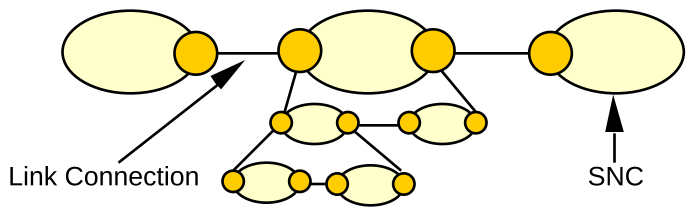

Fig. 3: Partitioning of a Nw Connection into SubNw. Connection

When the number of nodes and SNCs is large, this technique produces complex 2D representations of multiple layers, which are difficult to read and to navigate. In this context, VRML allows layer navigation to be performed in a natural way.

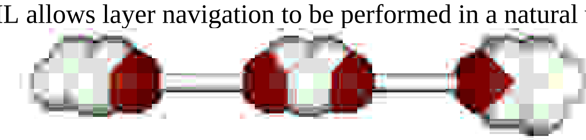

Fig. 4: VRML Representation of the Previous Figure

The figure above shows a VRML representation of SNCs. Subnetworks are represented as clouds whereas link connections are represented as pipes. When the camera is distant from the subnetworks, only the top level is displayed although the subnetworks contain further elements. Thanks to the VRML LOD tag, users can see the contained subnetworks and link connections by moving the camera towards a subnetwork. This operation is recursive hence contained subnetworks can be exploded by moving the camera closer towards them. Each element contains a hotlink, which allows users to jump to other pages containing detailed information about the element.
VRML can also be applied to represent hierarchical views of ATM networks. The PNNI routing protocol views the world as a collection of peer groups. At the bottom, a peer group is formed by an administrator logically drawing a circle around a collection of switches. These switches must be a set of devices which are connected like a graph. Given several peer groups connected to each other by border switches, one can build a hierarchical structure by drawing a circle around this collection of peer groups, and declare that to be a peer group, the parent of each of the constituent peer groups. As shown before, it is possible to query the PNNI interface in order to build a hierarchical view of an ATM network.

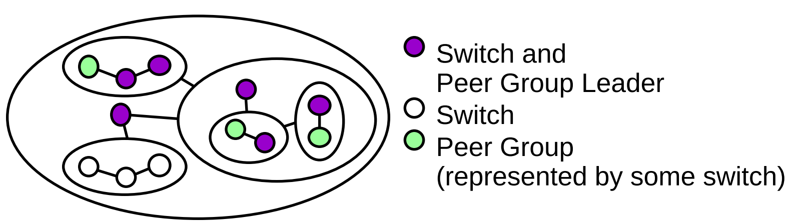

Fig. 5: PNNI Peer Group Hierarchy

The figure above represents how a collection of switches might be aggregated into a hierarchical topology, which can be represented in VRML as a sphere which represents a peer group containing the other peer groups, switches, or peer group leaders.

Fig. 6: VRML Representation of the PNNI Peer Group Hierarchy

The VRML representation is much more compact and expressive than the 2D one. Additionally it allows multiple peer groups to be nested and hence to be represented in a single picture. The LOD gives the user first an abstract view of the whole network and then allows a more detailed view of the peer group of interest by moving the camera close to it.

Compound Information View

Very seldom can all the potential information be represented in the same view since different users may be interested in different aspects of the same information. Additionally, in many cases the information to be represented is quite rich, so ways to "compress" such information have to be identified. This means that a representation has to be as clear as necessary and as rich as possible in order to depict most of the information in a single picture. Compound information means that several aspects of the same information have been combined to produce a simple representation that removes or hides any information that is irrelevant for a given representation. VRML can help in this respect because it allows a great amount of information to included in a single virtual world.

The following example shows how this concept has been applied to a real situation. The European ACTS project MISA (Management of Integrated SDH and ATM Networks), deals with the management of integrated ATM and SDH networks. Members of the MISA consortium, who are located in different European countries, provide access to resources relevant for the project. X.700 agents keep track of these resources: they contain object instances that represent locations, computers, and services provided. When this information is represented, it is important to show as much information as possible on a single screen to avoid having to walk through too many different screens. VRML facilitates this because it allows logic containment (country, location, computer, services) to be represented using real 3D containment and also because it allows views to be manipulated easily by moving the whole world and walking through it. The way to represent information, retrieved dynamically from the X.700 agents, is the following: a map of Europe has been wrapped over a 3D flat surface, partner locations are identified by building (boxes) located where the partner really is and labelled with the flag of the country. Services are depicted in a colour representing their state. Each element has an HTML anchor that links it to the corresponding resource details or that allow to jump to other VRML/HTML pages.

Agents contain all the information needed to represent the view stored inside object instances that contain topology information. When the VRML world is built, the topology instances are located and agents are no longer accessed dworld uring the manipulation. The object containment tree is used to produce the 3D containment whereas information about the precise element location, maps and anchors to detailed information, as well as links to VRML element views are contained in X.700 object attributes as part of the object instances. The following picture shows a simple VRML world representing MISA partners and the services they provide.

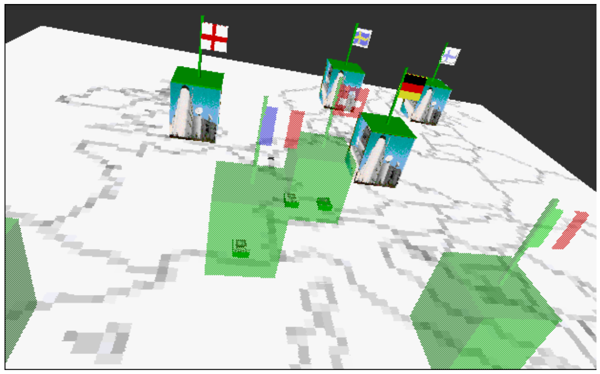

Fig. 7: MISA Partners and Services

The use of the LOD contributes to simplify the exploration of the world since the VRML viewer removes/adds objects to the view according to the current camera position. When the user is far away from a location where a MISA partner is located, a box with a flag on top of it is displayed. The closer the user approaches a box, the more details are shown. It is like navigating the containment tree with a camera: users do not have to enter boxes to see what is inside and also because it is possible to represent much information in one view without demanding users to click several times in order to arrive at the object of their interest, although this possibility is provided as well.

4 Liaison: VRML Extensions

The examples shown so far have been implemented as VRML extensions to Liaison. Liaison is based on software components called *droplets* [3] that have the ability to be replaced and added at runtime allowing the behaviour of the application that contains them to be modified and extended at runtime. Each of these VRML extensions have been implemented using droplets.

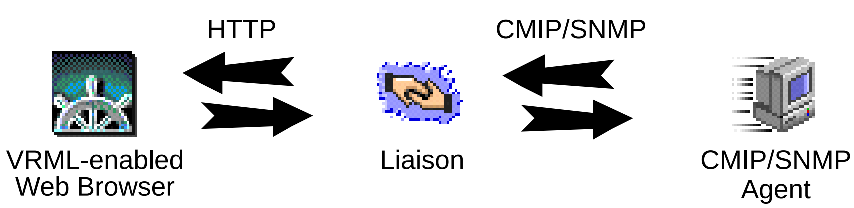

Fig. 8: Liaison's VRML Extensions

When an HTTP request is received, Liaison routes it to the droplet responsible for handling it. The later issues one or more CMIP/SNMP requests and builds an internal representation of the virtual world. Since VRML specifies the location of each ele-

ment, the droplet has to layout each element by assigning a 3D location to it. At this point the droplet returns an HTTP response and a file containing the VRML world whose type is `x-world/x-vrml`. The VRML-enabled Web browser (or the VRML viewer) receives the file and shows it properly according to the file type, which is VRML in this case. Once the user clicks on one of the HTML anchors contained in the VRML world, a request is issued and a new file is returned. It is possible to jump from HTML to VRML and vice-versa by using the file type, that in the case of HTML, is `text/html`.
For each object, the location and the name of the VRML file used to render the object are retrieved from the OSI agent. This file is basically a VRML file with certain parameters, replaced at runtime with their actual value, which represent translation coordinates, scale, rotation, anchors, etc. These parameters, which are not part of the VRML language, are defined as special strings contained in the VRML code. The following is a VRML code fragment which shows how these parameters are used:

```
WWWAnchor {
 name  "$anchor"                  # Object Name (URL)
 description "$title"             # Information about the object
 Translation { translation $X }  # Object position
 Texture2 { filename "$fn" }      # Texture to be mapped on the obj.
 Cube { }                          # Cube (default size)
} # WWWAnchor
```

The droplet responsible for VRML replaces the parameters at runtime with the actual values retrieved by the X.700 agents. This technique has additional advantages:

- if not all the objects have to be shown, the VRML location attribute can be set to null in order not to generate VRML code for those objects;
- it is possible to design an object using a sophisticated VRML editor and then to embed the parameters into the VRML code;
- the VRML file can be changed dynamically while the Liaison is running;
- the object instance information in the X.700 agent can be manipulated separately from the graphical VRML aspects;
- droplets greatly reduced the development time and code size since most of the services were already available and have been exploited thanks to the facilities provided by Liaison.

The only situation in which VRML is not suitable is whenever the management information changes frequently. VRML files, like HTML, are static, so they cannot be used in very dynamic situations because every time the file changes then the VRML viewer has to parse the file.

5 Conclusion

This paper showed how 3D visualisation can be used effectively for network management. It covered a novel technique developed to display management information using VRML.

The main characteristics of this technique are:

· Internet-ready, simple, and platform-independent based on VRML, which is a portable, compact, and widely established modeling language;
· ability to display 3D data using their native format instead of flattening them to 2D;
· ability to mix 2D and 3D data.

VRML has to be considered a novel and promising solution for Web-based 3D network management visualisation due to its wide range of appliance and its native compatibility with HTML. Its platform-independence, high flexibility and wide acceptance make VRML probably the most reasonable solution for 3D visualisation.

6 Acknowledgements

The VRML-based visualisation system described in this paper has been designed and implemented at the IBM Zurich Research Laboratory. The authors gratefully acknowledge partial funding by the European Commission, European ACTS project AC080 MISA, and by the Swiss Federal Office for Education and Science. Moreover, the authors would like to thank R. Akolk and D. Gantenbein for their suggestions and valuable discussions, in addition to users of *Webbin' CMIP*[4], who have greatly stimulated our work with all their comments and suggestions.

7 References

[1] Apple Computer Inc., *QuickDraw 3D Metafile Format (3DMF)*, October 1995.

[2] J. Case, M. Fedor, M. Schoffstall and J. Davin, *A Simple Management Protocol (SNMP)*, Network Working Group, RFC 1157, May 1990.

[3] L. Deri, *Droplets: Breaking Monolithic Applications Apart*, IBM Research Report RZ 2799, September 1995.

[4] L. Deri, *Surfin' Network Management Resources Across the Web*, Proc. of 2nd IEEE Workshop on Systems and Network Management, Toronto, June 1996.

[5] European Telecommunications Standard Institute, Telecommunications Management Network (TMN), *Generic Managed Object Class Library for the Network Level View*, I-ETS 300.653, DI/NA-043316, May 1996.

[5] ITU-T Recommendation G.805, *Generic functional Architecture of Transport Networks*, November 1995.

[7] Sun Microsystems, *The Java Programming Language*, Addison-Wesley, 1996.

[8] The ATM Forum, *Private Network-Network Interface Version 1.0 (PNNI 1.0)*, AF-PNNI-0055.000, March 1996.

[9] Silicon Graphics, *OpenInventor C++ Reference Manual*, Addison-Wesley, 1994.

[10] G. Bell, A. Parisi and M. Pesce, *The Virtual Reality Modeling Language (VRML)*, Version 1.0, May 1995.

[4] An on-line demo and a version available for public download (supported platforms: AIX, OS/2, MacOS, Linux, Win95/NT) can be found at http://misa.zurich.ibm.com/Webbin/.