

# VISUALIZATION SYSTEMS AND THE INTERNET

Ken Brodli  
School of Computing, University of Leeds  
kwb@comp.leeds.ac.uk

## ABSTRACT

The paper discusses how established dataflow visualization systems such as IRIS Explorer can be extended to take advantage of two new opportunities for visualization offered by the Internet.

The first is the use of the Web as a distributed computing environment in which visualization services can be provided. The techniques fall roughly into two classes: client-based systems, where execution of the visualization process takes place on the desktop, using software downloaded over the Web; and server-based systems, where execution of the visualization process takes place on a server machine remote from the desktop, but driven by instructions from the user. The paper describes the way these techniques have evolved over the past decade, and looks in particular how the traditional modular visualization environments such as IRIS Explorer can adapt to this new style of working.

The second is the use of the Internet for collaborative visualization. We describe the COVISA toolkit which extends IRIS Explorer from single-user system, to a powerful multi-user visualization system supporting distributed research teams.

**Keywords:** visualization, modular visualization environments, web-based visualization, collaborative visualization, IRIS Explorer.

## 1. INTRODUCTION

Visualization has become an increasingly important aspect of computational science and engineering, allowing us to comprehend the 'tidal wave' of data generated by increasingly powerful simulation and measurement technology. Essentially it is a transformation of numbers into pictures, allowing us to harness our highly developed visual senses in order to 'see the unseen'.

The history of visualization goes back a thousand years, but recent interest in visualization was sparked by an influential NSF report by McCormick, De Fanti and Brown [7]. This report argued that investment in supercomputer facilities in the US

needed to be complemented by similar investment in visualization facilities, if the real potential of high performance computing was to be realised. This report sparked the development of a number of powerful visualization systems, many still in use today.

These systems are of a family known as Modular Visualization Environments, or MVEs. They follow a common reference model, elegantly expressed by Haber and McNabb [3]. This model decomposes the visualization process into a number of stages (data generation, filter, map and render) that link together in a dataflow pipeline (see Figure 1). The filter stage selects the data of interest – a mathematics operation; the map stage chooses a visualization metaphor to convert from numbers to geometry – some call this science; and the render stage uses computer graphics techniques to turn geometry into an image – the engineering step.

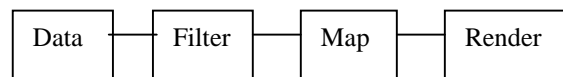
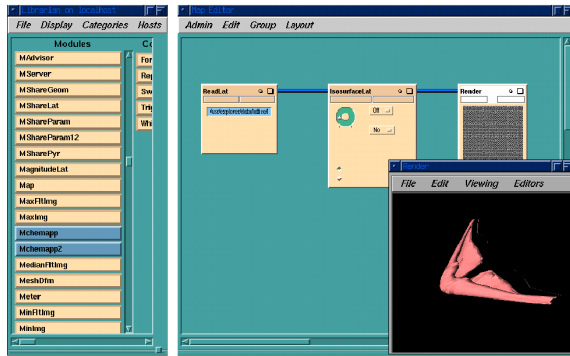


Figure 1 – Haber-McNabb Dataflow Pipeline

This model underlies the family of systems which originated with AVS [10] and continues today with IRIS Explorer [5,9] and IBM Open Data Explorer [4], as well as AVS itself. Figure 2 shows a screenshot of IRIS Explorer. To the left is the 'Librarian', which houses a set of modules for carrying out steps of the visualization process; the main window is the 'Map Editor' in which modules are connected together by the user into a dataflow network. The visual programming environment encourages experiments with different modules in order to create different visualizations. In the example shown, pressure data from an aircraft simulation is read in; an isosurface (of equal pressure value) is extracted as a set of triangles; and a render module generates a picture from these triangles.



**Figure 2 – IRIS Explorer Visualization System**

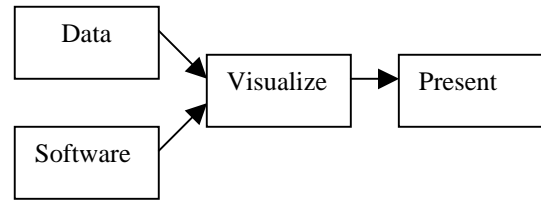
Over the past decade since these systems were first designed, the world of computing has changed dramatically. The emergence of the Web, first as a distributed information repository, later as a distributed computing environment, has transformed our outlook – from one which was centred on the desktop, to one in which global access to resources is taken for granted. This in turn has driven major advances in network performance – which in turn makes it feasible to work collaboratively in multi-site research teams.

In this paper, we look at how the dataflow visualization system can work in the Internet age. There are two main parts to the paper. In section 2, we look at Web -based visualization, and how an MVE can be used in either a client -side or a server -side approach. In section 3, we look at how an MVE can be extended to collaborative working. Section 4 looks forward to future opportunities.

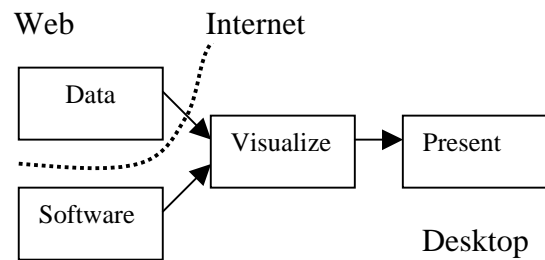
## 2. WEB-BASED VISUALIZATION

### 2.1 A Model for Web-based Visualization

In this section we shall build a reference model that will help to distinguish the different approaches to Web-based visualization. We start from the Haber - McNabb model of Figure 1, but simplify the model by combining the filter and map stages into a single Visualization stage. This has two inputs: the data to be visualized, and the software to create the visualization. Its output is passed to the Presentation stage, which is similar to that of the Haber -McNabb render. This is shown in Figure 3.

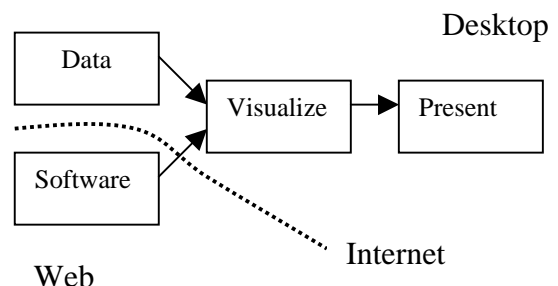


**Figure 3 – Simple Reference Model**

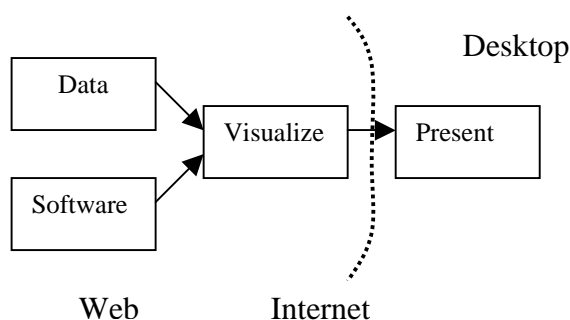


**Figure 4 – Web as a Data Repository**

In the original MVE concept, all processes would execute on the desktop. However, we can illustrate the different approaches to Web -based visualization by highlighting parts of the process which might occur remotely from the desktop, on the Web. In the first case shown in Figure 4, we are seeing the Web as a data repository and access the data to be visualized from a URL reference; execution remains on the desktop. In the second case, shown in Figure 5, we see the Web as a software repository and download the visualization software (typically as a Java applet). In the third case, shown in Figure 6, we see the Web as a computing facility where the visualization process executes and transmits geometry or images to the desktop.



**Figure 5 – Web as a Software Repository**



**Figure 6 – Web as a Computing Resource**

In the following sections, we see how MVEs can adapt successfully to each of these scenarios.

## 2.2 The Web as a Data Repository

The Web provides a global repository for information, allowing us to download data from governments, from companies, from research collaborators. A very simple extension of the MVE allows us to incorporate such data in a dataflow network.

In the case of IRIS Explorer, we have developed a data reader module that will access data, not from a local file, but from a URL reference. Immediately this allows a very simple form of collaborative working: one member of a research team can post results at a URL, and these can be visualized at any later time, by any other member of the research team.

A variation of this approach is to download the data as a link in a Web page. If the data file has an associated MIME -type, it is possible to cause the download to launch a visualization system to process the data. Indeed this was the approach taken by Ang et al [1] in the first demonstration of the Web for visualization, and subsequently used by Hibbard in Vis5d [11]. However this seems best suited to turnkey visualization systems, like Vis5d, where there is a unique visualization application to be launched, rather than a programmable system such as an MVE.

## 2.3 The Web as a Software Repository

The emergence of Java as a means of creating portable code, downloadable over the Web, offers fresh opportunities for visualization. An early idea was to develop visualization applets that can carry out particular types of visualization. For example, Michaels and Bailey [8] describe VizWiz, an applet for visualizing volume data. A difficulty with applets is the security restriction that forbids them from accessing data on the local machine. Hence if a user wishes to visualize their own data, this must

first be transferred to the applet host machine. Michaels and Bailey use the browser 'File Upload' facility to achieve this; Kee, as reported in Brodlie and Wood [2], used a Java application to fetch the data from a URL to the applet host machine.

In the case of MVEs, we can think of the software as being in two parts: the visual program and the environment. The visual program specifies the set of modules needed to create the visualization, and how they should be wired together. The environment provides the modules themselves, and the supporting software. Most MVEs have a scripting language as an alternative to the visual programming interface, and so it is possible to replace human actions of selecting and wiring modules with a pre-written script. In the case of IRIS Explorer, the scripting language is Skm.

This gives us the opportunity to develop repositories of scripts on the Web, each script being an MVE 'program'. By associating an agreed MIME -type with these scripts, we can cause the MVE to be launched on download of the script, and the corresponding dataflow network initiated. Again this work has been done in the case of IRIS Explorer. Yeo, as reported in Brodlie and Wood [2], built a demonstrator where an initial network was launched, and a sequence of further interactions were invoked on a step-by-step basis by interpreting further Skm commands.

## 2.4 The Web as a Computing Resource

In the final approach, the Internet divides the visualization and presentation components so that the construction of the visualization is performed as a server-side operation, with geometry or images being transmitted to the browser. One of the first instances of this approach is described by Wood et al [12]. Here IRIS Explorer is used to provide a Web -based air quality visualization service to the general public. The user enters data on a form, specifying the pollution data of interest, and how they would like to visualize it; this form is processed by a CGI script which invokes IRIS Explorer as a server process; VRML is generated and returned for viewing in the browser. This is illustrated in Figure 7.

# 3. COLLABORATIVE VISUALIZATION

## 3.1 A Model for Collaborative Visualization

Increasingly, the most challenging problems in science require the co -operation of a large,

multidisciplinary research team, with different people bringing different skills to the solution of the problem. Very likely, the research team will be distributed at different locations nationally, or indeed internationally.

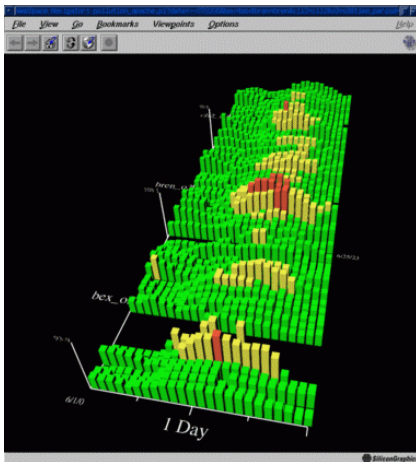
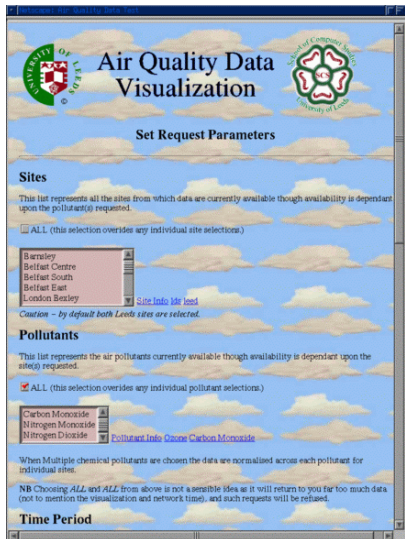


Figure 7 – Web-based Air Quality Service

The Haber -McNabb reference model extends very naturally to collaborative working. A very simple extension is shown in Figure 8. Here the data is supplied to each of the collaborators, who share access to the parameters which control the filter, map and render processes. The same network is used by each collaborator.

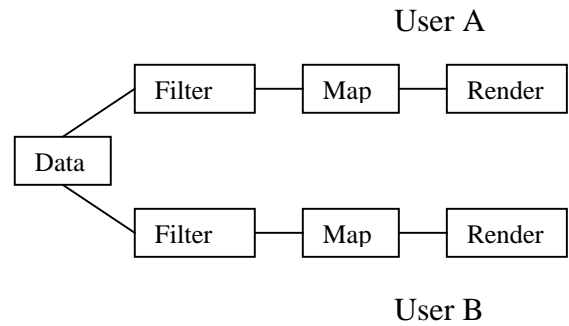


Figure 8 – Extending the Haber-McNabb Model – Two Users with Same Pipeline

A more powerful model arises when we allow each collaborator to have a different dataflow network. We can allow export of data from any collaborator at any point in the network, and corresponding import of data from another collaborator. We show this schematically in Figure 9.

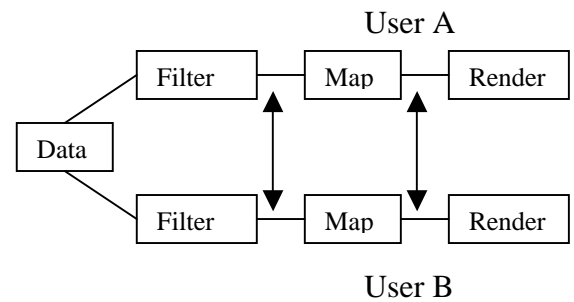


Figure 9 – Exchanging Data Between Collaborators – Users Here have Different Pipelines

This allows complex collaborative programs to be created, that interlink the individual networks of each collaborator. One typical instance is shown in Figure 10. Here one collaborator is responsible for creating the visualization, and then exports the geometry from the map process so that other collaborators can render the geometry locally.

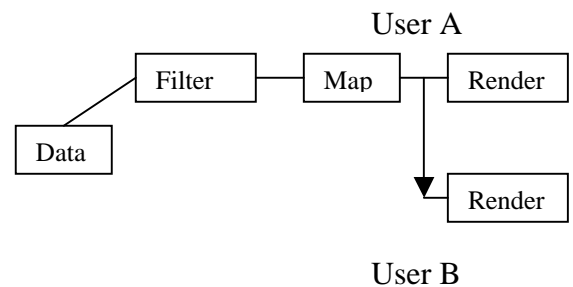


Figure 10 – Example Collaborative Network

### 3.2 IRIS Explorer COVISA Toolkit

This model has been implemented by Wood as the COVISA toolkit for IRIS Explorer, and it is available as part of the current release (release 5.0). The toolkit works as follows. A central COVISA server process manages the collaborative session, maintaining a list of users engaged in the session. To enter a collaborative session, a user launches an Mserver module which creates a link to the COVISA server, and downloads information on other users in the current session. To export some data, a user launches an 'Mshare' module and wires it into the dataflow network, just like any other Explorer module. The COVISA server becomes aware this has been done, and launches a complementary 'Mshare' module in the map editor of all other collaborators. If they wish to import this data, they can wire it in as they wish to their own network.

This is illustrated in Figures 11 and 12. In Figure 11, we see the initial map of Figure 2 extended by inclusion of an Mserver module, and the wiring in of two Mshare modules to export the data to another user, and to share control of the Isosurface module.

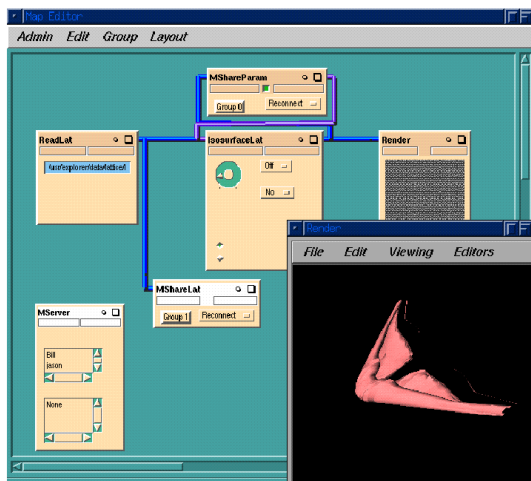


Figure 11 – Including Collaborative Modules

In Figure 12, we see the two sessions side-by-side. The lines show the transfer of data between sessions.

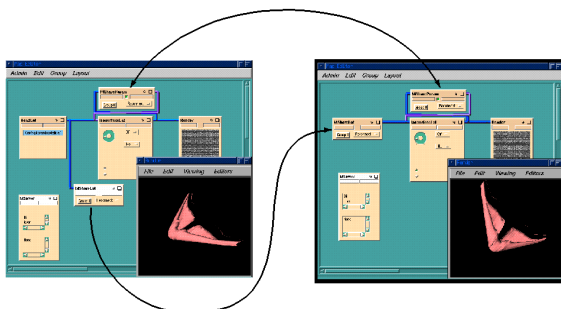


Figure 12 – A Collaborative Session

### 4. CONCLUSIONS AND FUTURE DIRECTIONS

We have seen how the traditional MVEs such as IRIS Explorer can sit comfortably in an Internet centred world. They can be used in a variety of Web-based scenarios, and they can be used to provide programmable collaborative sessions.

There remain many interesting avenues to explore. In Web-based visualization, the combination of downloadable Java software and MVEs is starting to be explored. Lovegrove [6] describes a prototype implementation where modules can be downloaded from software repositories, and connected together in a dataflow network. His system is collaborative, so that the downloaded modules are sent to each of a number of collaborators who share in the wiring process.

In collaborative visualization, most of the focus has been on 'data' visualization, where the data is pre-calculated. As interest in Grid computing escalates, so there is increasing interest in computational steering where the data is generated in real-time by a simulation process – and visualization is used as basis for steering decisions. We are starting to explore the opportunities for collaborative computational steering.

### ACKNOWLEDGEMENTS

I would like to thank Jason Wood who developed the COVISA toolkit as part of his PhD research, and who has also provided many of the ideas behind the Web-based visualization work.

### REFERENCES

- [1] Ang CS, Martin DC and Doyle, MD. Integrated control of distributed volume visualization through the World Wide Web. In Bergeron RD and Kaufman AE (editors) Proceedings of IEEE Visualization94, pp13-20, 1994.
- [2] Brodlied KW and Wood JD. Volume Graphics and the Internet, in Volume Graphics, edited by Min Chen, Arie E Kaufman and Roni Yagel, Springer, pp 317-331. ISBN 1852331925, 2000.
- [3] Haber RB and McNabb DA. Visualization Idioms : A Conceptual Model for Scientific Visualization Systems, in Visualization in Scientific Computing, edited by G.M. Nielson, B. Shriver and L.J. Rosenblum, pp74—93, IEEE, 1990.
- [4] IBM Open Data Explorer Web Site. See: <http://www.research.ibm.com/dx>. 2001.

[5] IRIS Explorer Centre of Excellence. Web Site. See: <http://www.comp.leeds.ac.uk/iecoe>. 2001.

[6] Lovegrove S. PhD Thesis, University of Leeds. 2001.

[7] McCormick B, DeFanti TA and Brown MD. Visualization in Scientific Computing, ACM SIGGRAPH Computer Graphics, 21 (6), 1987.

[8] Michaels C and Bailey M. VizWiz: A Java Applet for Interactive 3D Scientific Data Visualization on the Web, Proceedings of IEEE Visualization97, pp 261-268, ACM Press, 1997. (See also VizWiz Home Page, <http://www.sdsc.edu/vizwiz/vizwiz.html>).

[9] NAG Ltd Web Site. See: <http://www.nag.co.uk>. 2001.

[10] Upson C, Faulhaber T, Kamins D, Laidlaw D, Schlegel D, Vroom J, Gurwitz R, and van Dam A. The Application Visualization System : A Computational Environment for Scientific Visualization, IEEE Computer Graphics and Applications, 9 (4), 30--42, 1989.

[11] Vis5D. Vis5D Web Site. See: <http://www.ssec.wisc.edu/~billh/vis5d.html>, 2001.

[12] Wood JD, Brodlie KW and Wright H, Visualization over the WWW and its application to environmental data, Proceedings of IEEE Visualization96, pp81-92, 1996.