Structuring for Extensibility - Adapting the Past to Fit the Future

Julian Gallop, Christopher Cooper, Ian Johnson
CLRC Rutherford Appleton Laboratory,
Chilton, Didcot,
Oxfordshire, OX11 0QX, UK
+44 1235 821900
{J.R.Gallop | C.S.Cooper | I.J.Johnson}@rl.ac.uk

David Duce
Oxford Brookes University,
Gipsy Lane,
Oxford, OX3 0BP, UK
+44 1865 484379
daduce@brookes.ac.uk

Gordon Blair, Geoff Coulson, Tom Fitzpatrick
Computing Department,
Lancaster University,
Bailrigg, Lancaster,
LA1 4YR, UK
+44 1524 65201
{gordon | geoff | tf}@comp.lancs.ac.uk

ABSTRACT

In this paper, we describe a component-oriented approach to distributed cooperative working with particular reference to scientific visualization. The approach is being developed based on insights gained in previous work on dataflow architectures for distributed cooperative visualization and on open bindings for middleware to support adaptive multimedia systems.

Keywords

CSCW, adaptation, component technology, continuous media, scientific visualization, dataflow, heterogeneous environments.

INTRODUCTION

Computer-assisted scientific visualization is an essential tool in many scientific, engineering and medical applications. Whether from experiments, surveys or simulations, complex and voluminous data is relied upon and the human visual sense is a powerful means of gaining insight into hard problems. Over the past decade, a variety of visualization software systems has emerged and stabilised.

Visualization is also a collaborative activity. Scientists and engineers collaborate over data, experimental apparatus, insights and designs. Collaboration can take place between researchers in the field and in the laboratory and across disciplines; it can take place between dispersed groups of a multi-site organisation. The opportunities provided by computer networking mean that increasingly this is needing to take place at a distance. Distributed cooperative visualization offers opportunities for fostering new styles of collaboration as well as replacements for existing ones.

As a means of conveying essential insights between the participants, representations (visual or audio) of the task in hand complement the direct audio and visual communications between the participants. These representations can include diagrams, images, movies or sound. Depending on the application and situation, distributed visualization can be extremely demanding of the communication channel including data transfer of bulk data and moving images and audio.

Tele-immersion [7] is a further advance and allows dispersed users to inspect, control and change a virtual reality model, which may embody visual representations of scientific data. Changes of scene need to be communicated to all participants, yet the round trips must avoid time lags which are distracting to human perception.

Distributed continuous multimedia has usually been considered an area of expertise distinct from scientific visualization. As technology permits is becoming widely used in many ubiquitous applications such as entertainment, education and communication.

With electronic devices breaking out of the traditional desktop and server model and becoming designed for all kinds of situations, including the wall, the briefcase, the vehicle and the pocket, multimedia software needs to encompass a diversity of situations. Component technology may offer a way of building applications for different combinations of devices building on a single base of components and may also offer a way of combining major software components - a more coarse-grained approach

Not only is there diversity but also dynamic change as network connections drop in quality or become more congested. Dynamic change can also occur as a new participant joins an existing group session with a device with reduced capability (reduced screen or not capable of running all the software). Adaptation becomes important.

These two different strands of work, distributed cooperative visualization and the development of open bindings for component technology for adaptive continuous multimedia, are brought together in the Visual Beans project [12] (funded by the UK Engineering and Physical Sciences Research Council (EPSRC)). First of all, we present these two strands in turn, then we present the research issues which have arisen in the current project.

DISTRIBUTED COOPERATIVE VISUALIZATION

Several studies have been made in distributed cooperative visualization. A state of the art report [2] covered the field widely and one study [14] led to its being adopted within a recent release of a commercial visualization system, Iris Explorer [6].

One set of partners in the current project (CLRC RAL and Oxford Brookes) were responsible for technology research and support in MANICORAL, a European project concerned with distributed cooperative visualization. The project also involved a collaborating group of European geoscientists and specialists in human communication and cognition. It is described more fully in [3], a description of the reference model, and [4], implementation and experience. We consider a number of issues briefly here.

- Existing whiteboard software allowed previously prepared results to be viewed by everyone concerned. However collaborative working requires not only the sharing of results, but a sharing of the generation process. This is particularly needed when the purpose is to explore the data rather than to describe it. With this deeper sharing, participants can suggest alternative scenarios from their different skills and experiences, implement them, assess the results while in their virtual meeting and increase the chances of a better understood joint decision.
- In some situations, participants need to have precisely the same view and controls as each other (what you see is what I see). In others, although participants need to share the same application (otherwise why collaborate?), they need to have their own views of and controls on the common application model. This diversity may be enforced by unequal resource availability or encouraged by different individuals' expertise or styles of working.
- Even within one particular collaborating group (and the geoscientists are no exception to this), meeting styles may vary. One meeting may be in a seminar style, with one person presenting and many others listening and seeing. Another may be run by a group of peers who have control powers equal to each other with perhaps the exception of a chairperson who may exercise some perhaps limited floor control.
- If software for cooperative working requires major changes to application software, its chances of acceptance are much reduced. Instead such software needs to be seen as an extension of the normal working environment.

The method of application sharing was based on visualization software systems with a dataflow architecture (alternatively called Modular Visualization Environments). The user is provided with a large selection of modules and builds an application at run time by choosing modules and joining their instances with connections which specify a flow of data. Figure 1 shows an simple network, which nonetheless shows that data may branch out or be gathered in. Some connections are intended for bulk transfers - entire volumes of consistently structured data. Others may carry single float or integer data. Additional modules can be developed by the user and connected in a similar way. Examples of systems of this type are AVS/Express (used in MANICORAL), Iris Explorer and IBM's Open Data Explorer (OpenDX).

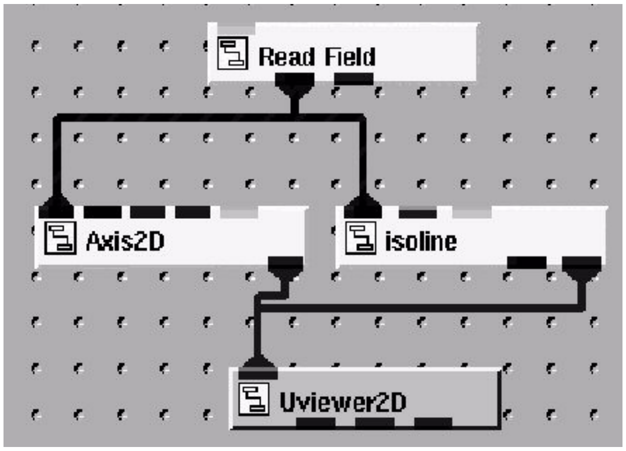
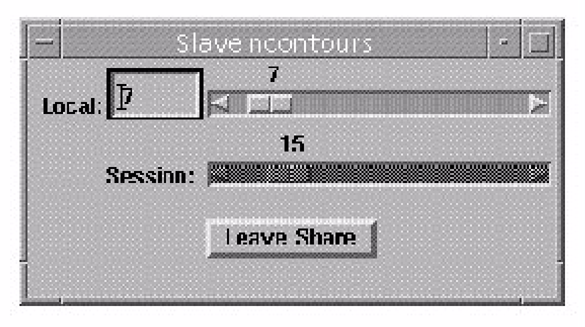


Figure 1. Simple dataflow visualization network

In MANICORAL, some modules were developed with the specific purpose of enabling cooperative working. One example is shared parameter control. This is based on the conventional single-user parameter control that is supplied in an unextended visualization system and extends it to allow multiple participants to control the parameter. Since the shared parameter control is implemented as a module like any other in the dataflow system and generates a float as its output, it can be located in the network anywhere that a float parameter is valid, which allows diverse possibilities. Each participant runs a visualization network and may include the shared parameter control module in it. If whole networks can easily be communicated to all participants, as was developed in the work described in [14], this replication is not onerous.

For the GUI, experimentation and improvement led to the interface shown in figure 2. The user not only needs to exercise local control (via text typed in or by a slider - both these are at the top in the figure) but needs to see the value that results from the actions of all participants, which is shown by the lower, greyed-out slider and which we refer to as the session value and which is delivered to the output port of the module. As in the figure, it would usually be expected that the two values would be different from each other. The shared parameter control modules are responsible for the policy for arbitrating the actions of the participants. A basic policy of “most recent action supercedes previous ones” was used. Alternative policies are possible and would alter the GUI but not the connections in the visualization network.

Knowledge of actual distribution mechanisms and machine addresses were separated out from the module and focused in a distribution server.



Local: 7

Session: 15

Figure 2. Shared parameter control in MANICORAL

Once this mechanism exists, any application based on this dataflow visualization system can incorporate any number of instances of these modules for cooperative working. Figure 3 shows a part of a visualization network. The two modules on the left of the figure are shared parameter control modules. All the others are conventional visualization modules. The output from each shared module is connected to two other modules.

Basing the work on a dataflow visualization system allowed the process of composition of an application to be flexible. The collaborative slider developed in MANICORAL could be attached to any module requiring a parameter, whatever the semantic level of that module.

However, the work in MANICORAL did not investigate continuous multimedia. This becomes necessary in visualization when dealing with time-based phenomena or indeed other phenomena which benefit from time-based presentation.

MANICORAL did not deal with communication between participants, which was implemented by existing public domain products. The resulting fragmented interface - where knowledge of who was present had to be sought in each of the GUIs for video, audio, whiteboard and application - was not addressed in this work.

It is instructive to compare the work on distributed cooperative visualization with other work. One approach is the collaborative zipper architecture described in [11]. The architecture of a collaborative application may have several logical layers. Some layers are replicated among the participants, others (generally the application semantics) are centralized. This has been likened in [11] to a zip being open (unzipped) or closed. Within its sphere of application, collaborative extensions to a dataflow visualization system can implement both replicated and centralized layers and the dividing line can be arbitrarily placed. In addition, such a system can implement more general collaborative architectures. An example of this is a situation where each participant may each introduce a dataset for comparison with others: here the zip would be open at both ends and closed in the middle.

Another useful comparison is with the idea of a collaboratory, in particular the Space Physics and Aeronomy Research Collaboratory (SPARC) [9], previously the Upper Atmospheric Research Collaboratory (UARC). This aims to provide a single desktop environment within which scientists collaborating at a distance can do their work, including shared whiteboard, shared chat, common access to views of data from multiple instruments and from multiple models. A collaboration needs all these aspects and test sessions in MANICORAL complemented public domain tools to provide a complete system. In addition, collaborative extensions to a dataflow visualization system are able to share the process of visualization, not just the results, and this may be a useful addition to a collaboratory’s toolset.

OPEN BINDINGS FOR ADAPTIVE CONTINUOUS MULTIMEDIA

In parallel to the work on distributed cooperative visualization, middleware has been studied as an important aspect of producing distributed applications. The development of OMG’s CORBA [8] has provided widespread industrial participation in a single set of specifications and has worked towards applications being able to mask out problems of distribution and diverse platforms and programming languages.

Traditionally the approach to using middleware has been to regard it as a black box and to build applications around it. In addition, the model of communication has been based on events and messages. The work at the University of Lancaster [5] has identified limitations of this approach when dealing with continuous multimedia. Further limitations ensue when dealing with widely diverse and varying CPU power, display characteristics and network performance - both the diversity and the variation widen when considering mobile and specialised devices.

The work described in [5] has developed an environment which:

- is based on CORBA.
- supplements the event-based communication mechanism in CORBA by introducing a stream interface designed to handle the production or consumption of a continuous media type, such as a stream of MPEG or WAV.

- adds to the closed bindings in CORBA the ability to make visible and control the logical associations between objects, thus providing an open binding mechanism (figure 4).

For instance, a distributed audio-visual application may make use of the stream interface. It may suffer variations in the network performance such as jitter and it may be necessary to use a buffering component to smooth this. When this has been detected by a monitoring component, the open binding allows an additional component to be inserted.

This environment (called TOAST) provides to the CORBA programmer a platform which offers a plug-and-play paradigm for constructing multimedia applications.

VISUAL BEANS - CONVERGENCE AND RATIONALE

The parallels between cooperative working based on dataflow visualization systems and the work on dataflow in adaptive multimedia suggests that convergence between component-based middleware and scientific visualization may be productive. The Lancaster work suggests that there may be better ways to build distributed cooperative visualization systems using component technology and the existing experience of dataflow systems, including the several GUIs that have been produced to control them, may be able to improve adaptation component software.

In addition, scientific visualization is a significant application of multimedia in its own right. Particularly, time-varying phenomena would require the use of continuous multimedia and such applications would benefit directly from the use of adaptation in TOAST.

ISSUES BEING EXPLORED

The Visual Beans project [12], which will run until 2002, has begun to explore these problems. Its aims are:

- To apply component technology and middleware to the construction of distributed cooperative visualization systems and to demonstrate their effectiveness.
- To develop strategies for adapting such systems in dynamically evolving heterogeneous environments.

In addition, it is intended to validate the approach through a series of application scenarios and to feed results to key standards bodies such as OMG.

A number of issues arise from this work:

- The MANICORAL work showed that a distributed cooperative application could be constructed at run time, given a suitable environment, such as a dataflow visualization system. We would expect component technology to bring further advantages, such as being able to migrate components to achieve a more effective distribution. Do these advantages actually occur in practice?
- Introducing continuous multimedia into a distributed cooperative application is not easy and raises issues of synchronisation between the participants and whether and when caching should be employed. Does the use of stream interfaces and open bindings as in TOAST allow this to be introduced?
- Distributed cooperative working involves making a number of design choices for different styles of cooperation, some of which ideally should be deferred and may need to be adjusted during a session. Does component technology - and in particular the adaptive component technology introduced by TOAST - provide a way of substituting alternative design choices (for instance different floor control policies)?
- In addition to the uses of adaptation already described, can adaptation also be effective for arbitrating between the demands of competing components and for managing the transition from single user to group use.
- Although several visualization systems employ the dataflow model, there is significant use of other patterns. An example is VisAD [13] which is gaining ground particularly in earth and atmosphere sciences. VisAD presents a rich class-based interface in and is written in Java and is based on a publish/subscribe model. How may such a system best be supported in a distributed component technology model intended to encourage plug and play and do the advantages still apply?
- Scientific bulk data types found in visualization differ from continuous multimedia in that losses are usually critical and the data can be regarded as non-continuous. Do the adaptation methods introduced by TOAST with the intention of supporting continuous multimedia also support delivery of these visualization data types?
- In dealing with VisAD, the grain of a component is likely to be coarse, whereas when supporting dataflow systems, the grain is likely to be fine. Is there a range of graininess most likely to be effective?
- When applying wrappers to build formal components, does one wrap to fit the programming model of the middleware or extend the middleware to accomodate multiple programming models?
- Does using design patterns provide a way of reasoning about these system design choices and predicting their effectiveness?

Although these questions have arisen out of applying these distributed computing and component technology to scientific visualization, we consider that they are in fact more generic.

ACKNOWLEDGMENTS

The authors acknowledge the support to the Visual Beans project provided by the (UK) Engineering and Physical Sciences Research Council (EPSRC) under grants GR/M81779 and GR/M82011.

REFERENCES

1. Blair, G.S., Stefani, J.B., "Open Distributed Processing and Multimedia", *Addison-Wesley*, 1997
2. Brodlie, K.W., Duce, D.A., Gallop, J.R., Wood, J.D., "Distributed Cooperative Visualization", *Eurographics '98 State of the Art Reports*, deSousa, A.A. and Hopgood, F.R.A. (Eds), pp27-50, 1998
3. Duce, D.A., Giorgetti, D., Cooper, C.S., Gallop, J.R., Johnson, I.J., Robinson, K., Seelig, C.D., "Reference Models for Distributed Cooperative Visualisation", in *Computer Graphics Forum*, 1998.
4. Duce, D.A., Gallop, J.R., Johnson, I.J., Robinson, K., Seelig, C.D., Cooper, C.S., "Distributed Cooperative Visualisation - the MANICORAL Approach", in *Eurographics UK Chapter Conference*, 1998.
5. Fitzpatrick, T., Blair, G.S., Coulson, G., Davies, N., Robin, P., "A Software Architecture for Adaptive Distributed Multimedia Systems*", IEE Proceedings on Software, Vol 145, No 5*, pp 163-171, Oct 1998
6. Iris Explorer Home Page `http://www.nag.co.uk/Welcome_IEC.html`
7. Leigh, J., Johnson, A., DeFanti, T., et al.," A Review of Tele-Immersive Applications in the CAVE Research Network" Proceedings of IEEE VR99, Houston TX, March 13-17, 1999
8. OMG, The Common Object Request Broker: Architecture and Specification, Revision 2.0, OMG 97-02-25, `http://www.omg.org/`
9. SPARC, SPARC background, `http://intel.si.umich.edu/sparc/background.htm`
10. Szyperski, C., "Component Software: Beyond Object-Oriented Programming*", Addison-Wesley*, 1998
11. Ter Hofte, H, "Working apart together: foundations for component groupware", Telematic Instituut Fundamental Research Series, No 001, ISBN 90-75176-14-7, 1998
12. Visual Beans Project Home page, `http://www.acu.rl.ac.uk/VisualBeans/`
13. VisAD Home Page, `http://www.ssec.wisc.edu/~billh/visad.html`
14. Wood, J.D., Brodlie, K.W., Wright, H., "Collaborative visualization", *Proceedings of IEEE Visualization 97*, pp 253-259, ACM Press 1997

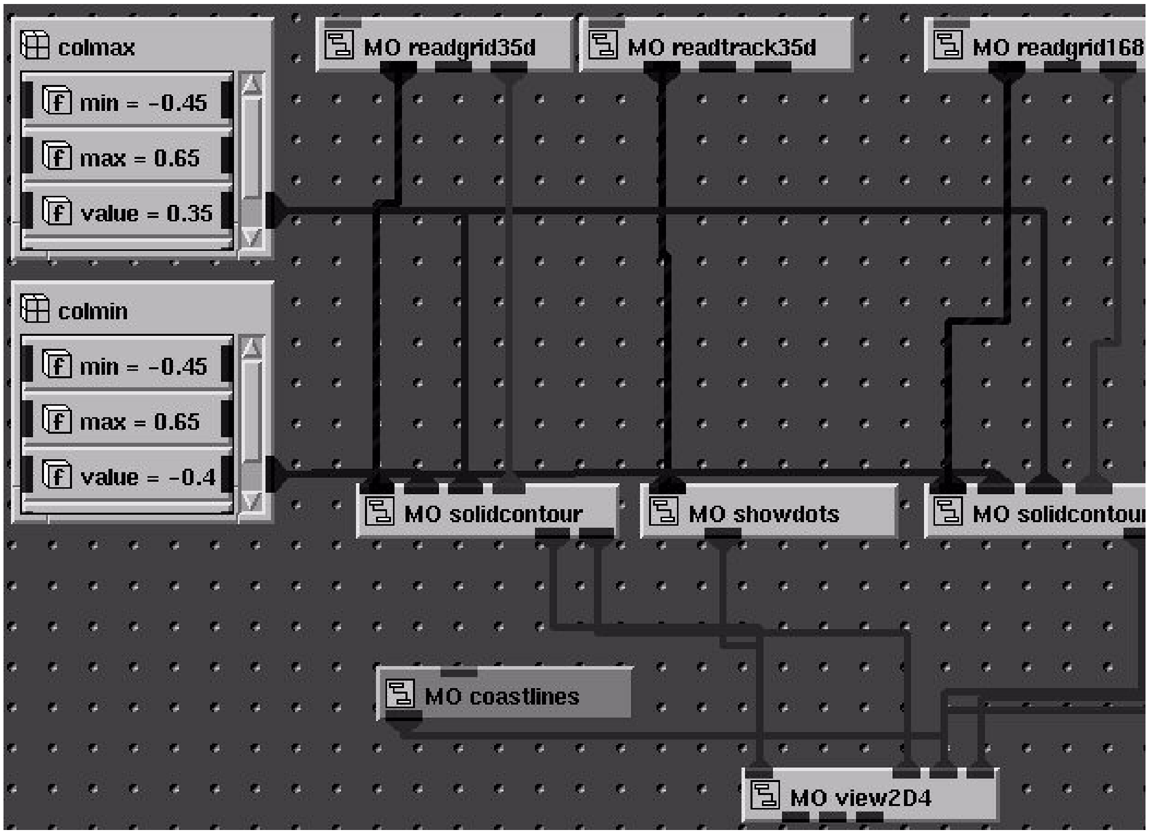


Figure 3. A section of a AVS/Express network with collaboration modules

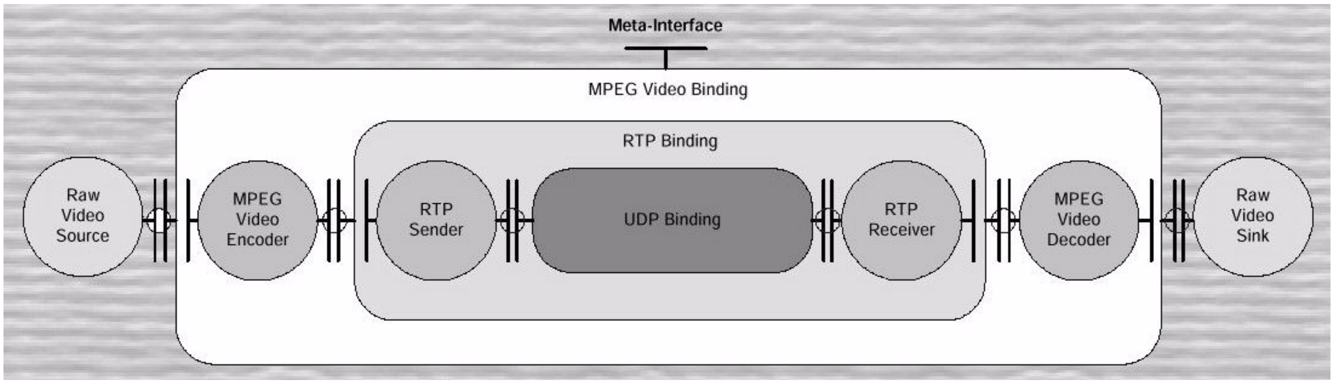


Figure 4. Open bindings for adaptation