

# XML for Visualization

K.W. Brodlie, J. Wood  
University of Leeds, Leeds, UK

D.A. Duce  
Oxford Brookes University, Oxford, UK

J.R. Gallop  
CCLRC Rutherford Appleton Laboratory, Didcot, UK

D. Gavaghan, M. Giles  
University of Oxford, Oxford, UK

S. Hague, J. Walton  
Nag Ltd, Oxford, UK

M. Rudgyard  
Streamline Computing Ltd, Warwick, UK

B. Collins, J. Ibbotson, A. Knox  
IBM UK Ltd, Hursley Park, Winchester, UK

## Abstract

**This paper describes work in progress in the gViz project. We suggest four areas in which XML should be applicable: data representation, visualization presentation, visualization application description, and audit trail for project history. We present initial thoughts on these areas.**

## 1. INTRODUCTION

Visualization is a key component for understanding large-scale simulations and observations. The modern era of visualization was initiated by the influential NSF Report, 'Visualization in Scientific Computing' [1]. This stimulated the development of a number of visualization software systems, most of which remain in widespread use. Some of these are general-purpose, notably the family of modular dataflow systems that includes IRIS Explorer [2], IBM Open Visualization Data Explorer [3] and AVS [4]; others are aimed at more specific applications or computing environments, such as pV3 [5] which is targeted at distributed memory parallel computing environments. In many visualization applications, particularly those involving an online simulation, the compute requirement is significant. Most systems are therefore designed to allow part of the processing to be done remotely, with just the final rendering at the desktop. As early as 1989, Haber and McNabb demonstrated this concept of 'remote visualization' at SIGGRAPH [6]. However the mechanism for running remote processes is typically simplistic and insecure. More recently remote visualization via the Web has been developed. For example, Wood et al. showed how IRIS Explorer can be used to provide a server-side Web-based visualization service for air quality data [7]. A number of studies of collaborative visualization have also been made - for example, the EU MANICORAL project [8], the EPSRC Visual Beans project [9] and the EPSRC COVISA project [10]. In the last case, this has resulted in a commercially available extension to IRIS Explorer. For a survey of this area, see [11].

As the move to a Grid model of distributed computing and data management gathers speed and the possibilities for collaboration over the Grid take shape, there is a need to review the effectiveness of current visualization systems in a Grid context, to identify the barriers to effective use and to specify necessary developments. The new requirements coming from the Grid include:

- Increased emphasis on distributed visualization, with part of the processing on the Grid, part on the desktop - this requires efficient and secure means of data transfer, as well as distributed algorithms to process the visualization data across large numbers of processors. An important aspect of this is computational steering, where real-time visualization of a simulation is used to improve parameter settings.

- Need to integrate with emerging Grid technologies, such as the Grid Security Infrastructure (GSI) to handle authentication, GridFTP to transfer large data files, and other Globus facilities. The advent of the Open Grid Services Architecture (OGSA) [12, 13] gives rise to the need to integrate with web technologies including XML and Web Services.
- Need to extend the models for collaboration: members of a team may have different preferred systems (present solutions require all users to work with the same system); collaboration may be asynchronous, ie occur over time (present solutions support only synchronous collaboration); and members may wish to switch between individual, synchronous and asynchronous working.

This paper describes work in progress in an EPSRC funded project entitled "Visualization Middleware for e-Science", the "gViz" project, which began in August 2002. The aim of the project is to research and develop visualization middleware for e-Science, integrating existing visualization systems with evolving Grid technologies, investigating compression techniques for transmitting large volumes of data, and exploring the use of XML for future developments in visualization. The project has close links with e-Science projects at the Oxford e-Science Centre, a case study from the Computational Biology Laboratory at the University of Leeds [14, 15], and the direct involvement of three industrial companies, to ensure the practical relevance of the work.

In this paper we focus on the potential roles for XML languages in visualization.

## 2. INTEGRATION OF VISUALIZATION AND WEB TECHNOLOGIES

XML has made a significant impact in many areas of computing, from e-business to mathematics. It is being increasingly used as the middle tier of client-server interfaces where its power and flexibility makes it ideal for middleware (for example, SOAP and related Web Services developments in W3C). In computer graphics, XML has been used in standards for web 2D and 3D graphics (W3C's SVG and Web3D's X3D - the successor to VRML).

In its simplest form, a visualization system may be thought of as a black box transforming a data source into a rendered visual presentation. Visualization systems in general do not perform a single fixed transformation from data source to visual presentation, but enable users to select (from a fixed repertoire) or construct (from some family of routines or modules) transformations appropriate to their data. There is thus a need to distinguish between the visualization system *per se* and particular transformations performed on particular data sources. We will refer to the latter as *visualization applications*. Visualization applications are executed by visualization systems.

Visualization can be used in different ways. Bergeron has given a classification [16]:

- *exploratory visualization* (undirected search): we do not know what we are looking for, visualization may help us to understand the nature of the data by demonstrating patterns in the data;
- *analytical visualization* (directed search): the process followed when we know what we are looking for in the data, visualization helps to determine if it is there;
- *descriptive visualization*: used when the phenomenon represented in the data is known, but the user needs to present a clear visual verification of this phenomenon (usually to others).

From this analysis of the ways in which visualization can be used, there also arises a requirement to document the usage of visualization in a particular context. A scientist exploring a new data set, will, for example, wish to record the visualizations generated from the data set, the applications used, the values of control parameters used, etc. Brodlie et al. developed this idea in the GRASPARC project [17]. In some contexts, this requirement may be a very formal audit requirement, for example in forensic work or to support a patent application.

Four points at which XML languages can be used in support of visualization emerge from this discussion:

- to represent the data input to the visualization system or data exchanged between distributed portions of a visualization application;
- to represent the visual presentation generated by the visualization system;
- to represent the visualization application applied to the data;

- to represent the history of a project, involving a set of collaborators, in which a variety of visualization applications are applied to a collection of data sources, generating a set of visualizations. Visualization applications may be used by individuals within the project or by groups of collaborators.

The following sections discuss each of these points in turn.

## 2.1 Representation of data

The evolution of data formats within different scientific disciplines is an interesting phenomenon. It seems true in general that new types of measuring instruments or new types of analysis software are introduced initially with proprietary (to the manufacturer, or research group) data formats. As usage becomes more widespread or interest grows in comparing data from one instrument with data obtained from another, so data formats become more standardized, perhaps not in a formal sense, but at least in an informal agreement amongst a significant user community to gather and exchange data based on a particular format. To minimize storage requirements and no doubt for other reasons also, data formats are typically binary data formats.

Most existing visualization systems use their own proprietary formats for describing the data to be visualized. See Brodlie et al. [18] for a review of the data models and formats provided by a range of visualization systems. Over a period of time, translation modules have been incorporated to enable data formats which are commonly used in the scientific community to be imported into visualization systems.

There are a number of projects exploring the use of XML to define commonly used data formats, including HDF [19], netCDF [20] and XDF [21]. There are two approaches to representing scientific data using XML:

- represent a description of the structure of the data using XML, with a reference to a binary (or other non-XML representation) of the data values;
- represent the structure and values using XML.

The first approach has the disadvantage that XML tools cannot be used to access the data values themselves. The second has the disadvantage that the markup can become extremely verbose. BinX [22] is an interesting example of the hybrid XML descriptor/ binary data file approach. BinX, currently work in progress, aims to provide the ability to describe the physical representation and the overall structure of binary data files. There is interest in binary representations of XML, for example the WAP Forum have proposed a tokenized binary representation for XML (as used in WML) [23]. However, this still suffers from other limitations of XML such as the fact that all XML documents are trees and this is not a convenient basis for representing, say, multidimensional arrays of data values. Although multidimensional arrays can be represented as trees, this is a poor representation for commonly required representations such as extracting a slice of the array or the values along a diagonal.

Schema namespaces provide a convenient abstraction for defining datatypes, and as Atkinson et al point out [13] there is a need for the types commonly used for e-Science applications to be defined in this way, in order to avoid a plethora of standards and types. The challenge is to find agreement. A way forward here might be to start with an uncontroversial, lowest-common-denominator set of schema fragments, then try to build consensus from there.

Data formats will be required to function with a wide range of data access mechanisms, for example data repositories, output from data mining services and output from the execution of mathematical models.

## 2.2 Representation of visual presentations

Visual presentations take a variety of forms depending on the spatial dimensionality of the presentation (usually 2D or 3D) and the temporal dimension, i.e. whether the presentation changes over time (animation). Visual presentations can be represented in a variety of ways, for example: 2D raster images, 2D vector graphics, 3D surfaces, 3D volume renderings. Time dependent behaviour can be captured at the image level, using movie formats such as MPEG, or at a higher vector/surface primitive level with animation formats.

There are examples of existing presentation formats that use XML, for example, SVG [24, 25] (W3C's Recommendation for 2D Scalable Vector Graphics) and X3D [26] (an XML representation of VRML [27]). Visualization systems have also been developed that use XML in this way. Lovegrove and Brodlie, for example, used VRML to represent collaborative worlds in which multiple participants could explore a visualization of a data set [28]. Other examples of the use of VRML include [7, 29].

SVG is being used in a variety of visualization applications, to capture visual presentations. An interesting feature of SVG in this context is the facility the standard provides for representing time-dependent behaviour (time animation). A number of papers at the SVG Open Conference in July 2002 [30] described applications of SVG in visualization, primarily in cartography and geographical information in a broad sense.

### 2.3 Representation of visualization applications

Using XML for representing data sources and visual presentations is, in a sense, conventional, though there is work to be done to identify generic languages that visualization systems should aim to support in order to meet the needs of a broad range of applications.

A less-conventional application of XML is to capture a description of the visualization application, or transformation, that the visualization system performs. This is interesting for a variety of reasons. When recording the history of visualizations in a project, the visualization application used to generate a visualization may be an essential component of the metadata associated with the visualization. Another reason concerns the place of visualization within the Open Grid Services Architecture (OGSA). As Foster et al. write [12] "A basic premise of OGSA is that everything is represented by a *service*: a network enabled entity that provides some capability through the exchange of messages". Key notions in OGSA are service type, service creation (of an instance of a specified type via a factory) and service lifetime management. There is an issue of the level of granularity at which visualization is represented in this model. One approach could be to view a particular visualization system as a service type and a visualization application as a subtype. A description of the visualization application could be passed to a factory in order to create an instance of a visualization system configured to provide that application.

Many existing visualization systems already have an internal interface separating the user interface from the 'engine' of the system. In IRIS Explorer, for example, this is the 'map' file which describes the module connections and parameter settings. As a practical first step in the gViz project the IRIS Explorer 'map' file will be expressed in XML syntax - giving immediate access to the growing set of XML technology, such as parsers, correctness validators, meta languages for resource description and so on. This will allow us to move the description of a visualization application between different computer systems. For example, by parsing the XML description, we can automatically generate a thin-client interface to IRIS Explorer, allowing access to visualization services from a Web browser. This provides a three tier approach: thin Web client; middle visualization layer (IRIS Explorer); and lower simulation layer on Grid resources.

There is an example of the use of XML to capture the description of a process in W3C's Scalable Vector Graphics (SVG) Recommendation. SVG provides a set of 15 types of filters. These may be combined into a dataflow filter network which is applied to primitives after rendering but before display. This is very reminiscent of the type of dataflow network used to represent visualization applications in a modular visualization environment. Figure 1 shows an example of a filter network, and figure 2 shows the visual effect of each stage of the network and the resulting effect of the whole network (the rendered duck over the final "merge" box).



```

<feComposite in="SourceGraphic" in2="specOut2" result="litPaint"
  operator="arithmetic" k1="0" k2="1" k3="1" k4="0"/>
<feMerge>
  <feMergeNode in="offsetBlur"/>
  <feMergeNode in="litPaint"/>
</feMerge>

```

The approach is essentially to name the arcs connecting filters in the network, represent filters by elements in the SVG XML language, and represent connections between filters by attributes (such as "in", "in1", and "result") on the elements which reference the corresponding named arc.

Viewing the visualization system as a service type and the application as a subtype is a pragmatic approach. From a more philosophical viewpoint, the visualization application itself is a service type and the visualization system on which it is realized is almost an irrelevant detail. In this view, the visualization application is a system independent description, which can be implemented in different visualization systems (including future systems not yet invented).

This higher level view of a visualization application service also accords well with emerging work on distributed business processes and workflow [31]. This focuses on the need to define service interfaces for visualization in such a way that they can be composed to form e-Science applications, for example, by composing simulation and visualization components. Achieving computational steering and collaborative visualization within this service composition view are interesting challenges, firstly because of the need for a feedback path from visualization to simulation, and secondly the need to identify groups of users permitted to join a collaborative session either at the start of a session or whilst a session is active.

This work has wider potential for the future. For example, the thin-client approach opens the way to Grid access from mobile devices which incorporate Web browsers - 'mobile computational steering'. In addition, XML descriptions would allow interworking of different visualization systems, and they would also allow storage of intermediate stages of an investigation, with application to asynchronous collaboration. Finally the XML description approach might lead eventually to a proposal for a 'standard' visualization language. One of the issues that has to be addressed is the lifetime of visualization application descriptions. Is the lifetime limited to the lifetime of the instance of the visualization system with which a visualization is created (far too limiting), to the lifetime of the type of the visualization system with which it is created (for example, if there is still an IRIS Explorer system somewhere in the world, a description relative to IRIS Explorer can be used to reproduce the visualization), or can descriptions persist and have meaning beyond the lifetime of the particular visualization system used to create a visualization in the first place? In the overall context of information curation, the latter view, that the description is in some sense independent of the type of system with which a particular visualization was created, is coming to be regarded as highly desirable.

Looking to the future, an approach of this kind may be important in the context of the Semantic Web/Grid, for example in order to answer queries of the form, "What kind of visualization might be effective for data with these characteristics for this kind of task?".

As a final remark in this section, we note that visualizations and visualization applications are often developed incrementally. There seems to be some merit in an approach to visualization description that acknowledges this explicitly.

## 2.4 Representation of project history

Although this area lies outside the immediate scope of the gViz project, it is an important area, and given the growing interest in the Semantic Grid or Knowledge Grid, it seems to us that this is a potentially interesting dimension to explore, building on the OGSA framework, visualization application descriptions and the history tree notion developed in GRASPARC as a way to characterise how visualizations have been produced in a collaborative project context. In general scientific work involves a mixture of individual and group working and asynchronous as well as synchronous modes of collaboration. History mechanisms need to encompass this. One potentially interesting avenue to explore is the extent to which the execution trace of a workflow description language can be used to provide a basis for a history mechanism. It has also been pointed out to us by one of the referees that the approach taken in the W3C Annotea project [32] (use of RDF to describe

annotations and XPointer to locate them in an annotated document) might be a useful approach to adding notes to the visualization application.

### 3. SUMMARY

This paper has presented the EPSRC-funded gViz project. We have explored a number of ways in which XML is important in visualization and have presented some initial thoughts on how these might be developed in the course of the project.

### ACKNOWLEDGEMENTS

Financial support from the EPSRC e-Science programme for the gViz project (including grants GR/R96224/01 and GR/R96217/01) is gratefully acknowledged. We would also like to thank the anonymous referees for their comments.

### REFERENCES

- [1] McCormick, B.H., de Fanti T. and Brown M.D. (1987) Visualization in Scientific Computing. *Computer Graphics*, **21**(6).
- [2] IRIS Explorer [http://www.nag.co.uk/Welcome\\_IEC.html](http://www.nag.co.uk/Welcome_IEC.html).
- [3] IBM Open Visualization Data Explorer <http://www.research.ibm.com/dx>.
- [4] AVS <http://www.avs.com>.
- [5] pV3 <http://raphael.mit.edu/pv3/pv3.html>.
- [6] Haber, R.B. and McNabb, D.A. (1990) Eliminating Distance In Scientific Computing: An Experiment In Televisualization, *International Journal of Supercomputer Applications*, **14**(4), pp 71-89.
- [7] Wood, J.D., Brodlie K.W. and Wright, H. (1996) Visualization over the World Wide Web and its application to environmental data, *Proceedings of IEEE Visualization96 Conference*, edited by R.Yagel and G.M. Nielson, pp 81–86, ACM Press.
- [8] Duce, D.A., Giorgetti, D., Cooper, C.S., Gallop, J.R., Johnson, I.J., Robinson, K., and Seelig, C.D. (1998) Reference Models for Distributed Collaborative Visualization, *Computer Graphics Forum*, **17**(4), pp. 219-233.
- [9] Gallop, J., Cooper, C., Johnson, I., Duce, D., Blair, G., Coulson, G., Fitzpatrick, T. (2000) Structuring for Extensibility - Adapting the Past to Fit the Future: *Proceedings of CBG2000, the CSCW2000 workshop on Component-Based Groupware*, December 2, 2000, Philadelphia, USA. ISBN 90-75176-24-4
- [10] Wood, J.D., Wright, H. and Brodlie, K.W. (1997) Collaborative Visualization, *Proceedings of IEEE Visualization 1997 Conference*, edited by R. Yagel and H.Hagen, pp 253–260, ACM Press.
- [11] Brodlie, K.W., Duce, D.A., Gallop, J.R. and Wood, J.D. (1998) Distributed Cooperative Visualization, *Eurographics 98 conference State of the Art Report*, ISSN 1017-4656.
- [12] Foster, I., Kesselman, C., Nick, J.M., Tuecke, S. (2002) The Physiology of the Grid. Available at <http://www.globus.org/research/papers/ogsa.pdf> (Version 6/22/2002).
- [13] Atkinson, M. et al. (2002) Rationale for Developing with the Open Grid Services Architecture in the UK e-Science Core Programme. (<http://aspen.ucs.indiana.edu/CCPEwebresource/c633gridatkinson/c633FINALUK-ArchitectureRationale+RoadMap17July02v0.1.pdf>).
- [14] Biktashev, V.N. and Holden, A.V. (2001) Characterisation of Patterned Irregularity in Locally Interacting, Spatially Extended Systems: Ventricular Fibrillation. *Chaos*, **11**, pp. 653-665.
- [15] Clayton, R.H. and Holden, A.V. (2002) A method to quantify the dynamics and complexity of re-entry in computational models of ventricular fibrillation, *Physics in Medicine and Biology*, **47**, pp. 225-239.
- [16] Bergeron, D. (1993) Visualization reference modles (panel session position statement). In G.M. Nielson and D. Bergeron, editors, *Proceedings of Visualization '93*, IEEE Computer Science Press.
- [17] Brodlie, K.W., Brankin, L. Banecki, G. Gay, A. Poon A. and Wright, N. (1993) GRASPARC - a problem solving environment integrating computation and visualization. In G.M. Nielson and D. Bergon (editors), *Proceedings of Visualization '93*, IEEE Computer Science Press.
- [18] Brodlie, K.W., Gallop, J.R., Grant A.J., Haswell, J., Hewitt, W.T., Larkin, S., Lilley C.C., Morphet, H. Townend, A. Wood, J. and Wright H. (1995) Review of Visualization Systems, AGOCG Technical Report No. 9 (Second Edition). Available at <http://www.agocg.ac.uk/reports/visual/vissyst/rvsrepor.htm>.
- [19] The HDF5 Information Page, <http://hdf.ncsa.uiuc.edu/HDF5/XML/>.
- [20] Caron, J. et al. The netCDF Markup Language (NcML), <http://www.scd.ucar.edu/luca/netCDF>.
- [21] The eXtensible Data Format (XDF) Homepage, [http://xml.gsfc.nasa.gov/XDF/XDF\\_home.html](http://xml.gsfc.nasa.gov/XDF/XDF_home.html).
- [22] Westhead, M. and Bull, M. Representing Sciencic Data on the Grid with BinX - Binary XML Description Language. See <http://www.epcc.ed.ac.uk/~gridserve/WP5/Binx/sci-data-with-binx.pdf> and the OGSA-DAI

project Web site <http://www.epcc.ed.ac.uk/~gridserve/>.

[23] WAP Binary XML Content Format, W3C Note 24, June 1999. Available at <http://www.w3.org/TR/wbxml/>.

[24] Duce, D.A., Herman, I. and Hopgood, F.R.A. (2002) Web 2D Graphics File Formats, *Computer Graphics Forum*, **21**(1), pp. 43-64.

[25] SVG, *Scalable Vector Graphics (SVG) 1.0 Specification*, W3C Recommendation, September 2001. Available at <http://www.w3.org/TR/SVG/>.

[26] X3D. See <http://www.web3d.org/x3d.html>.

[27] Virtual Reality Modelling Language (VRML) (1997), ISO/IEC 14472-1.

[28] Lovegrove, S. and Brodli K. (1998) Collaborative Research within a Sustainable Community: Interactive Multiuser VRML and Visualization. *Proceedings of the Eurographics UK Chapter Conference*. ISBN 0-952 1097-7-8.

[29] Engel, K.D., Westermann, R. and Ertl, T. (1999) Isosurface Extraction Techniques for Web-based Volume Visualization, *IEEE Visualization '99*, pp. 139-146.

[30] SVG Open Conference Proceedings. See <http://www.svgopen.org/>

[31] Automating Business Processes and Transactions in Web Services. See <http://www-106.ibm.com/developerworks/webservices/library/ws-autobp/>.

[32] The W3C Annotea Project. See <http://www.w3.org/2001/Annotea/>.

© University of Leeds, Oxford Brookes University, CCLRC Rutherford Appleton Laboratory, University of Oxford, Nag Ltd, Streamline Computing Ltd, IBM UK Ltd. 2002

IRIS is a registered trademark of Silicon Graphics, Inc.

IBM is a registered trademark of International Business Machines Corporation.