Visualization Reference Models

David M. Butler, Chairman
Limit Point Systems, Inc. at Sandia National Laboratories, California
butler@ca.sandia.gov

James C. Almond
University of Texas System Center for High Performance Computing
j.almond@hermes.chpc.utexas.edu

R. Daniel Bergeron
Computer Science Department, University of New Hampshire
rdb@bergeron.cs.unh.edu

Ken W. Brodlie
School of Computer Studies, University of Leeds, Leeds, UK
kwb@scs.leeds.ac.uk

Robert B. Haber
Department of Theoretical & Applied Mechanics
University of Illinois at Urbana-Champaign
haber@csrd.uiuc.edu

Introduction

Although the majority of visualization research has focused on specific visualization problems, there has recently been increasing interest in trying to understand visualization in general - in trying to define high level visualization reference models[1-8]. Some of the issues surrounding reference models include:

Do we understand visualization in general? Is visualization a general process with many special applications or a collection of unique, unrelated, specific techniques?

Should there be a single reference model for visualization in general, or many domain specific models, or a hierarchy of models?

Should a model address correctness, robustness and efficiency?

Should the scope of a visualization model include the several traditional domains on which visualization relies: visual perception, computer-human interface, computer graphics, etc.? In particular, should the scope include traditional computer graphics only, the mapping from application to graphics, or application domain only?

Who would use a reference model: users, providers, developers, standards bodies? What would they use the model for: choosing and learning visualization techniques, evaluating systems, designing systems, defining standards?

What is the connection between reference models and standards? Is it time to start discussing visualization standards? What would be standardized? What would be the benefit? What would be the cost? Would standards promote progress or inhibit it?

The panelists will address these and other questions as they review the state of the art in visualization reference models and explore current and future research issues.

The panelists are all actively involved with reference models and/or standards activities:

David M. Butler is the principle consultant at Limit Point Systems, Inc., a consulting firm specializing in the design of systems for processing scientific, engineering and medical data. For the last several years he has worked with Sandia National Laboratories on integrated environments for scientific computation, visualization and data management.

James C. Almond is the Director of the University of Texas System Center for High Performance Computing in Austin Texas. He has been a member of the DIN and ISO CGI and CGM standards committees and a member of the ANSI computer graphics reference model committee. His current technical interests include distributed scientific visualization techniques, methods for the hierarchical storage and access of scientific information, and the formal definition of data structures for object-oriented and parallel programming methods.

1070-2385/93 $3.00 © 1993 IEEE

Robert B. Haber is a Professor of Theoretical & Applied Mechanics at the University of Illinois at Urbana-Champaign. He received his Ph.D. in Civil Engineering from Cornell University, where he worked in the Program for Computer Graphics from 1975 - 1980. He has been an active member of the visualization community, heading visualization-related research projects at the National Center for Supercomputing Applications and the Center for Supercomputing Research & Development.

R. Daniel Bergeron is a Professor of Computer Science at the University of New Hampshire. He was active in the Siggraph Graphics Standards Planning Committee as co-chair of the Core Graphics subcommittee (1977) and the Input Task Force (1979). Professor Bergeron's current research is focused on the integration of multi-dimensional scientific data visualization techniques into a comprehensive scientific database environment.

Ken Brodlie is a Senior Lecturer in the School of Computer Studies at the University of Leeds, and Deputy Head of the Division of Computer Science. He has had a long involvement with international standards for computer graphics, and presently chairs the group looking at the revision of GKS. His interest in scientific visualization began in the 1970s with an involvement in the NAG Graphics Library, and has extended to research into more general problem solving environments. This work has been focused on the GRASPARC project (funded by the UK SERC and DTI) which is looking at the integration of computation, visualization and data management.

Position statement for David M. Butler: reference models and practical necessity

I imagine that the popular belief among visualization software developers is that a reference model for visualization is unnecessary. It's at best something to keep trailing-edge committees busy and at worst a positive impediment to progress in a rapidly evolving research area. I hold that a robust, abstract, *mathematical* model of visualization is essential to producing the coming generation of complex, highly visual software systems that a multitude of scientific and engineering applications demand.

There is no longer any doubt as to the utility of visualizing scientific data. The last several years have seen a variety of stunning achievements in the field of scientific visualization; stills and animations that have impressed scientists and laymen alike. These widely heralded successes have created a strong demand in the scientific user community for ubiquitous, easy to use visualization software integrated into every field of investigation. But the software responsible for these visualizations has been anything but ubiquitous and easy to use. The most successful visualizations have usually been produced by teams of visualization experts using (or developing) expensive special purpose codes. Recently, general purpose visualization environments have appeared, but these still require expert knowledge to use effectively. How will the visualization community deliver to the practicing scientist the world of ubiquitous visualization that our past successes (read advertisements) have promised?

Furthermore, the demand does not stop at the bounds of the traditional scientific world. The global engineering and manufacturing community is rapidly evolving standards and software to automate information handling throughout the product life cycle. The advent of affordable high performance computing is currently making traditional scientific simulation techniques such as computational mechanics and fluid dynamics an integral part of the product design activity. Hence, "scientific" visualization will soon become a mainstream industrial activity. The traditional approach to visualization software will clearly not be able to satisfy this demand. So how do we proceed?

Obviously, we proceed with the best software development practices available and today there is little question that the best practice is object-oriented. The object-oriented paradigm (OOP) emphasizes the construction of reusable software abstractions called classes. The two main mechanisms of reuse are inheritance, which allows a programmer to easily create a new, special purpose class from an existing more general class, and dynamic typing, which allows the new, more specialized class to be used by any existing "client" program that was written to accept the more general class. Both these mechanisms require the more general, more abstract class to be identified, designed and coded first, before the more specialized classes. Abstract classes that have not yet been coded can not be inherited, nor can programs be written to use them. The usual inductive method of discovering useful software abstractions after implementing many special cases carries a real penalty in the object-oriented approach: not only do inheritance and dynamic typing not help in implementing the special cases, the special cases must actually be rewritten to "reuse" the abstraction once it's discovered.

The conclusion is immediate: to effectively develop visualization software we need detailed reference models which define the abstractions to use in designing class libraries and application frameworks. It is interesting to note that this role is quite different than the after-the-fact standards role traditionally allotted to reference models.

In the object-oriented world, a good abstract reference model drives a standard by utility, not by legislation.

To the immediate objection "but where do we find such models?" I have two responses. The first is, we must consciously look for useful *abstractions*, rather than just enumerating yet more special cases. The second is that when seeking abstractions, mathematics is the tool of choice. In particular, most scientific disciplines are already mathematized. Focusing software design on the mathematical formalisms of application domains rather than on the data formats of existing implementations is a large step in the right direction.

Position statement for James C. Almond: reference models and the essence of information

In our world of information, the technological and political walls of yesteryear are tumbling down. As distance is overcome by communications technology, the interchange of information is playing a major role in eliminating boundaries. In communication as in computing, all essential activities can be seen as operations performed on informational entities; we create them, store and access them, transmit, share, compare, transform and visualize them. Even the process of human visual perception involves a mental (if subliminal) operation on an image --- an informational entity typically in some pixel format. As the information industry matures and becomes increasingly globalized, our traditional focus on computing must expand to include an enhanced understanding of these informational operations, based on a coherent technology for the uniform representation of informational entities.

The computing industry has not done well at this. Traditional systems are based on strings of bits called "files". Their spontaneously defined structure and encoding typically make them application specific, often in an effort to distinguish a proprietary product. Descriptive information as to information content, structure or encoding is typically lacking in both the file system and in the file structure itself. As a result, efforts to establish interoperability have often degenerated into an arbitrary enumeration of specific cases.

There is increased interest in enhancing file based storage systems by providing descriptive metadata, stored separately from the bits of the file in the interest of accessibility. Current interest in Abstract Data Types (ADT's), and Object Oriented (O-O) techniques are further indications of progress. While O-O methods offer an applicable technology for systems development, their effective use for numerically intensive scientific computations is still controversial. Clearly however, the potential of such technologies can be fulfilled only if a coherent framework can be established for the definition of informational entities based on improved fundamental principles.

Let us consider three concepts as a possible basis for the development of such a framework:

1. Let the Informational Essence (IE) of an entity possibly specified as an object class or an abstract data type (ADT), uniquely define its information content, and thus the domain of its functional semantics. This concept offers a logical basis for a uniform taxonomy of information, and supports hierarchical definitions of IE's composed of other IE's.
2. Let the structure of an entity define a mapping of its logical elements to the one-dimensional address spaces used by computer storage media. (Regular structures often use implicit mappings, such as the Cartesian product for the elements of matrices and arrays. Irregular structures must resort to more explicit mechanisms, such as used for sparse matrices and database elements.)
3. Let the encoding specify how individual elements are represented in the binary language of the computer.

These three concepts are conceptually orthogonal. For example, the semantics of a square matrix with real valued elements are not affected by its storage as a structure of rows, columns, diagonals, blocks, or indexed structures, nor are either semantics or structure affected by the representation of the elements in any specific format. Such clear factorization of orthogonal concepts has the potential for reducing complexity by promoting the systematic definition of objects and ADT's.

The definition of IE's themselves is fundamental, but difficult due to the fact that entities tend to be domain specific. The use of reference models is a powerful method for resolving processes into discrete stages operating on precisely defined IE's, and for discovering common concepts which can be re-used among domains. (Unfortunately perhaps, the concept of a reference model is also used for describing alternatives at a given stage or level, e.g. for characterizing all possible visual representations of a given multidimensional set of discrete data.)

Some principles involved in such process-oriented reference models include:

1. the discretization of a process into stages which produce precise changes in the IE of entities,
2. precisely defined interfaces between stages,
3. the capture of information at these interfaces into metafiles, containing entities of precisely defined IE,
4. the consideration of informational entropy produced by (possibly irreversible) transformations,

5. the possibility that a stage may interface directly to multiple stage instances at a higher level (fan-in) or at a lower level (fan-out), resulting in tree-like models.

Reference models are being used to describe computer graphics, visualization techniques, and text processing. Their comparison reveals some interesting identities, and offers insights as to fundamental principles for the identification of appropriate IE's.

Position statement for Robert B. Haber: visualization reference models

The branches of engineering and science in which visualization is an important tool are many and diverse. Nonetheless, there is a good deal of commonality in the visualization operations and data representations used in these fields. The abstract data model based on fiber bundles, proposed by Butler and Pendley and elaborated by Haber, Lucas and Collins, helps to reveal this common structure. While it is possible to identify some visualization applications that are not well-served by this model, they are far outnumbered by those that do. Therefore, the notion of a cross-disciplinary reference model for visualization has merit.

A widely-accepted reference model would benefit both developers and users of visualization systems. It would provide a structure for the design of extensible, object-oriented visualization systems and provide a common terminology and conceptual framework for describing visualization data and operations. It would also simplify the design of interfaces between applications and visualization systems. Lastly, a well-designed reference model appears to be a prerequisite to the development of standards for visualization systems.

The reference model should be based on a high-level abstraction that is not application specific, such as the fiber bundle model. Instances of the reference model could then be elaborated in a hierarchical, object-oriented fashion to address the requirements of specific disciplines. Similarly, specific instances of the model can be developed to address trade-offs between robustness and efficiency. However, the robustness of the high-level abstraction that underlies the reference model is critical, since it determines the range of applicability of the model and, therefore, its ultimate viability. Accuracy is a critical concern in scientific and engineering applications. One advantage of the fiber bundle model is that it suggests natural metrics for quantifying error in both independent and dependent data sets.

An effective object-oriented model for visualization can be constructed from an abstract data model and an abstract description of operations on data. While one could logically include the application itself in the model, I would suggest for pragmatic reasons that the reference model be limited to operations that describe mappings (in either direction) between application data and graphics data and between graphics data and images. However, this implies that the reference data model must include suitable models for application data sets.

The emergence of a visualization standard based on a well-defined reference model would certainly be beneficial. However, the standards-making process is an uncertain enterprise and it is probably the case that no existing reference model is sufficiently mature for adaptation in its present form. To simplify the problem of developing a standard, it might be desirable to concentrate initially on a standard data description. This would support the free exchange of data between applications and visualization systems, and defer the thornier problem of constructing a general framework for defining visualization operations.

Position statement for R. Daniel Bergeron: goals for a data visualization reference model

Principal goals of any reference model

A reference model provides a conceptual framework describing a complex multi-faceted problem domain. A good reference model should describe the problem at a sufficiently high level of abstraction that both software designers and users are able to understand and use the common principles that underlie the domain. At the same time, it must be sufficiently concrete that both designers and users are able to translate their specific application-oriented needs to the concepts and structures defined by the reference model.

Thus, the principal goal of a reference model is to achieve conceptual unification and simplicity. Other goals, such as correctness, robustness and efficiency are clearly secondary and should follow naturally if an appropriate conceptual model is discovered. In other words, a good reference model should lead to implementations that are complete, reliable, and efficient, but should not be so detailed as to prescribe (or proscribe) any particular implementation technology. Similarly, a good reference model can be an important influence on the development of standards, and probably should be a pre-requisite step for any standards effort, but it is not and should not be treated as a standards activity. Standards should be defined once a domain has achieved a level of maturity such that it is possible to gain reasonable consensus on what could be termed "accepted practices"; development of a reference model can help a field to mature to that point.

Reference model for visualization

The development of a reference model for any domain forces researchers and users to analyze the knowledge and practices of that domain in order to identify basic principles and concepts that may not be obvious in advance. Identifying common principles is more difficult when the domain is both new and multi-disciplinary. This is especially true of data visualization which has the added complication that it is usually driven by specific mission-oriented goals of the particular application.

The problems in understanding the visualization domain are further exacerbated by the diversity of both applications and user goals that fall under the general rubric of "visualization". In order to define a reference model for visualization, we must first define the desired scope for that model. In effect, that means we must rigorously define what we mean by "visualization" -- or at least what we are willing to include in the reference model.

The visualization task

To have any hope of being truly effective, a visualization reference model must identify the goals of the visualization as determined by the user, rather than just the visualization task itself. User-oriented goals can be roughly divided into three categories: "descriptive visualization", "analytical visualization" and "exploratory visualization". Descriptive visualization is used when the phenomena represented in the data is known, but the user needs to present a clear visual verification of this phenomena (usually to others). Analytical visualization (directed search) is the process we follow when we know what we are looking for in the data; visualization helps to determine whether it is there. Exploratory visualization (undirected search) is necessary when we do not know what we are looking for; visualization may help us understand the nature of the data by demonstrating patterns in that data. These broad goals share some subgoals (and, therefore, can share some tasks), but they also have their own unique characteristics that demand unique facilities. A visualization reference model should be able to encompass the needs of each of these major goals.

Data representation

When we think of the user's goals, it is clear that a major component of the visualization task must include steps for data extraction and transformation as well as visualization. Consequently, a visualization reference model must address issues of data representation and transformation. In effect, the visualization reference model must incorporate a model for scientific data that is able to represent a wide range of application data in a comprehensive and cohesive fashion. We need a data model that goes well beyond a data format standard, although that is also necessary. A data model must be powerful enough to support meaningful data analysis and transformation operations that can be readily shared across users and even application domains.

Summary

Scientific data visualization has developed rapidly into an independent, multi-disciplinary field. Its explosive growth and its enormous potential are contributing to random, uncoordinated, and unfocused development. It could be extremely valuable to initiate an effort to develop a visualization reference model aimed at identifying basic principles that underlie the field. Such a model should be user and task driven and should include as a significant component the development of an appropriate data model. Especially in the early stages the reference model should concentrate on high level conceptual functionality and avoid focusing too heavily on standards requirements.

Position statement for Ken W. Brodlie: visualization, reference models, and standards

Visualization is now recognized as a key aspect of scientific computing. Although still an active area of research, the subject has reached a level of maturity that it is now timely to ask the question: should standards for visualization be developed?

Computer graphics standards

In the mid-1970s, computer graphics had reached a position where there was general agreement on the need for standards. There were many competing graphics systems (but largely of one type - FORTRAN subroutine libraries), and it was impossible to transfer applications between systems without considerable editing. Thus a contouring routine, for example, written in terms of one graphics library could not be readily used with a different underlying library.

Hence the now well-chronicled graphics standards activities began. But it is interesting to reflect on the approach taken. It was decided just to "bite off" a small part (2D schematic graphics) in the hope that sufficient understanding and international agreement could be reached *quickly*, so as to produce a first standard. This was GKS - published in 1985. Other areas (3D graphics, metafiles, device interfaces) were separated out for later processing. All have now been published.

A consequence of this piecemeal approach is some difficulty in interworking. For example, GKS assumes a metafile to be an "audit trail" of a session; the metafile standard, CGM, is more concerned with the storage of a set of independent pictures. PHIGS introduces some concepts, even for 2D graphics, which are not present in GKS.

Work is now underway on a second generation of graphics standards, including a revision of GKS (nearly complete) and a new standard aimed at a wider multimedia scope. However, between the two generations, an important consolidation effort took place in the creation of a Computer Graphics Reference Model [9]. This defines computer graphics in terms of five abstract levels or "environments", and identifies operations that can be performed on data elements in each environment. The distinction between audit trail and picture capture metafiles, that was blurred in GKS, is well defined in the reference model. An annex to the reference model standard describes each existing standard in terms of the model, and highlights discrepancies.

The reference model has been an important guiding influence in GKS revision, and hopefully it will ensure the next generation of graphics standards form a coherent and logical set.

Visualization

One can argue that the state of the art in visualization today is not unlike that of computer graphics twenty years ago. It is a lively and popular subject; there are a growing number of software products (though the variety is perhaps greater than for graphics- subroutine libraries, command driven packages, menu-driven systems and visual programming application builders); there is a growing interest in a theoretical basis for the subject, giving it some maturity; there is a need from the user community to be able to port applications between systems.

Thus it is timely to consider the need for standardization. But before undertaking such a Herculean task, it is important to address a number of issues:

1. Is there a will? Standards are only worth developing when there is a strong push from the user community. There must be a will to have a set of international standards whose validated implementation will be insisted upon in procurements.
2. How to develop? ISO graphics standards have been developed by committee work, involving a wide spread of interests and nations. Consensus has required compromises. The resulting standard is then passed to the community for implementation. The alternative approach is to adopt a product already in the market place and seen as a *defacto* standard.
3. Role of a Reference Model. Do we begin the standardization process with a reference model, and then construct visualization standards which relate to that model? For example, the reference model might identify filtering, mapping and rendering as "layers" in the model; then individual standards would be constructed in each layer, with rules for transferring data between layers to ensure compatibility. The alternative view (as taken by the graphics community in the 1970s) would be to identify the niche where standardization is most needed (say, a standard data model?) and progressively develop other standards. Once a body of experience has grown up, then the reference model is defined and the existing standards retro-fitted as best as can be done. Obviously the former approach is better - if we have the necessary knowledge at this stage, and if we can do it quickly enough.

References

1. Bergeron, R.G. and Grinstein, G.G. "A reference model for the visualization of multi-dimension data", Proc Eurographics'89, pp 303-399.
2. Brodlie et al. (eds) Scientific Visualization, Springer-Verlag, 1992
3. Butler, D.M. and Pendley, M.H. ; "A visualization model based on the mathematics of fiber bundles"; Computers in Physics 3(5) 45 (1989)
4. Butler, D.M. and Bryson, S. ; "Vector-Bundle Classes Form Powerful Tool for Scientific Visualization", Computers in Physics, Nov-Dec 1992, pp 576-584.
5. Butler D.M. and Hansen, C., "Visualization'91 workshop report: scientific visualization environments", ACM Computer Graphics v.26 n.3 1992, pp 213-216.
6. 4th Eurographics Workshop on Visualization in Scientifc Computing, April 21-23, 1993.
7. Haber, R. B. and McNabb, D.A., "Visualization idioms: a conceptual model for scientific visualization systems", in Visualization in Scientific Computing, ed. Nielson, G.M., Shriver, B., Rosenblum, L.J., 1990, pp 74-93.
8. Haber, R.B., Lucas, B., and Collins, N, "A data model for scientific visualization with provisions for regular and irregular grids", Proc. Visualization'91, pp 298-305.
9. ISO/IEC Information technology - Computer graphics - Computer Graphics Reference Model, ISO/IEC 11072:1992, 1992.