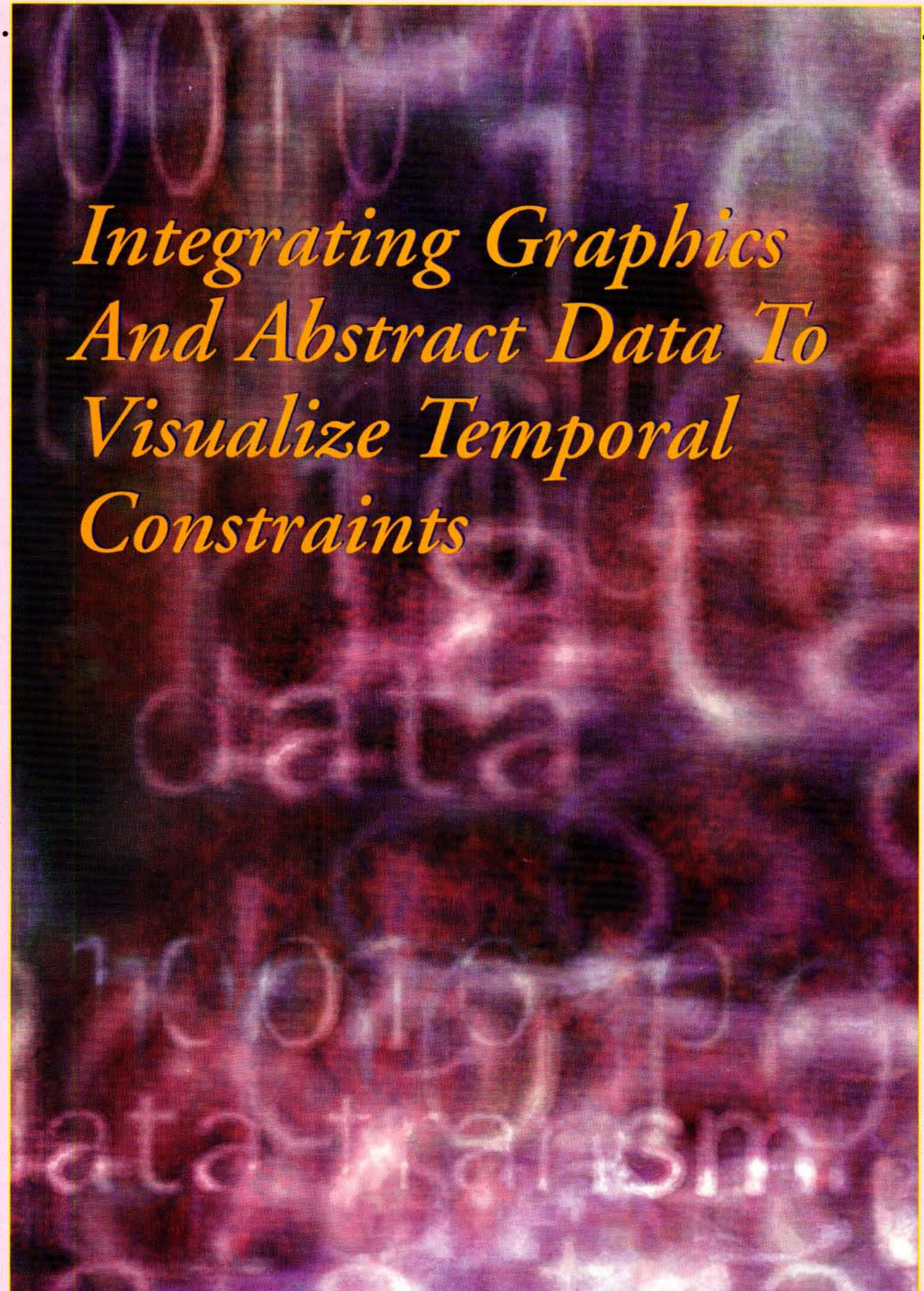
Integrating Graphics And Abstract Data To Visualize Temporal Constraints



Don Bishop ©1997 Artville, LLC

Louis Hoebel, William Lorensen, and Ken Martin[1]

General Electric Corporate Research and Development
Schenectady, NY 12301
{hoebel | lorensen | martink}@crd.ge.com

Traditionally, visualization is the transformation of data into information that can be rendered using computer graphics techniques. Visualization combines techniques and representations from computer graphics, computer vision, and image processing. In distributed and hierarchical operations and processes, such as for command and control or logistics and planning, visualization is the central mechanism for communicating the state of the situation and operations. The major challenge of visualization is to filter, tailor, and present the information in compact forms that can be efficiently created and displayed. In contrast to many visualization problems, these domains have what seems to be an overwhelming variety and quantity of information. We address aspects of this problem with the graphical visualization of abstract temporal information in a concrete spatio-temporal framework.

Time is nature's way of keeping everything from happening all at once.
—Richard P. Feynman, in lecture

Introduction

How do we show everything *not* "happening all at once" but still in a unified framework and within a unified presentation? How do we add this and other dimensions to data?

The initial exploration comes from the concept of an information lens. Figure 1 is an example of an information lens: a tissue lens generated from computed tomography (CT) data. In this mode of visualization, *context* is provided by one view of the data—the skin image, an isosurface in graphics terminology—and *detail* by another view—the skeletal view, in which the CT volume is used to generate isosurfaces of the skin and bone.[2] The tissue lens moves freely about the body image and displays the specific and requested skeletal data in the context of the complete body image [Schroeder et al. 1997].

[1]Ken Martin performed this work while at GE CRD. He now works at Kitware and can be reached at *http://www.kitware.com*.

[2]The skin is *clipped* away by a sphere to reveal the bone, and a sphere cutter is used to reveal the internal structure on the surface of the cut volume.

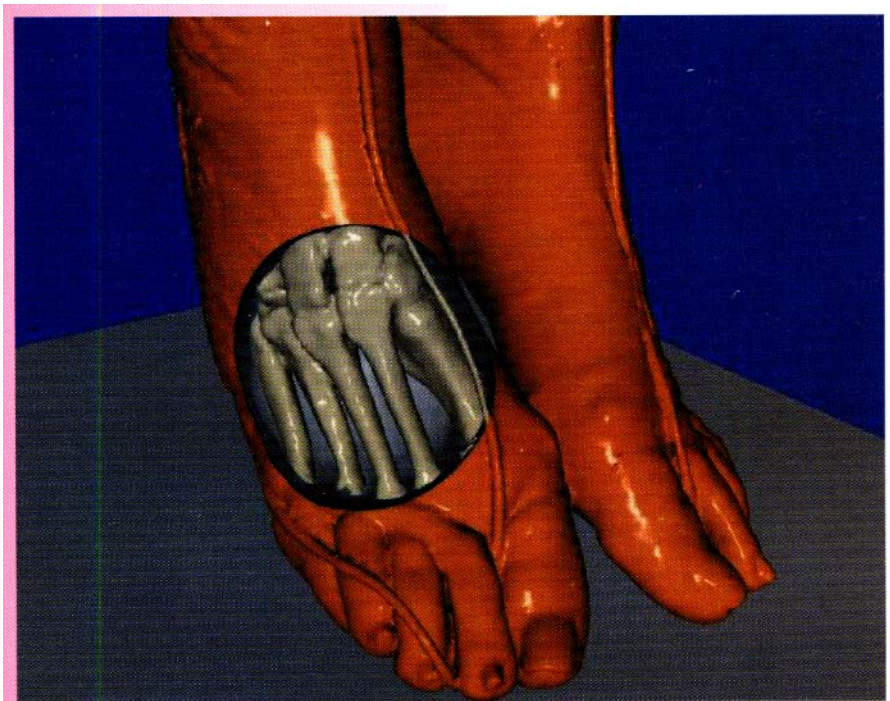


Figure 1. Complex visualization combining clipping, cutting, and volume rendering to produce a tissue lens, a special type of information lens.

The problem to which we applied this tissue lens concept involved military planning. The motivation is represented conceptually in Figure 2. In this case, the motivation is the use of a synchronization matrix to construct in time the major movements of a detailed and precise special unit or small unit operations. Planning occurs in time, whereas viewing occurs in space. The initial idea was to construct a lens that would allow viewing of temporal data through a clipped region of a map, thus providing spatial context of where and when events might occur.

In this article, we address the issue of visualizing abstract data (nonscientific data) within the context of integrating a temporal constraint reasoner and a computer graphics toolkit. Tachyon is a constraint-based temporal reasoner developed at General Electric Corporate Research and Development (GE CRD) [Arthur and Stillman 1992, Arthur and Stillman 1993, Tachyon 1994]. The graphic rendering comes from Visualization Toolkit (vtk), also developed at GE CRD by a separate group [Schroeder et al. 1997].

Visualization Toolkit

Visualization Toolkit is an emerging and publicly available standard for visualization. Features germane to the domains mentioned include a general nested transformation; a variety of one- , two- , and three-dimensional dataset representations; and a demand-driven dataflow pipeline. The pipeline is a collection of visualization objects that implement visualization algorithms. The algorithms are connected by specifying input-output relations. Data are pulled through the pipeline by explicit requests by application analysis or rendering operations. For applications that need integrated image processing with visualization, vtk provides an extensive suite of image processing objects that can be intermixed with conventional visualization datasets.

The image processing is multithreaded to accelerate the processing of large image datasets. Image processing is streaming, allowing multiple pipeline passes over large datasets that cannot be kept memory resident. Vtk contains sets of rendering objects, but other rendering interfaces such as Open Inventor can also be used. Vtk wrappers provide a multiprogramming language interface to vtk. Although vtk is written in C++, features exist to automatically create other language interfaces to the vtk objects.

Vtk wrappers provide a bridge between the application programming language and the C++ vtk objects using wrappers for Java, Tcl/Tk, and Python. This means developers can work in C++ or in a target language of their subsystem and not have to learn a less familiar lan-

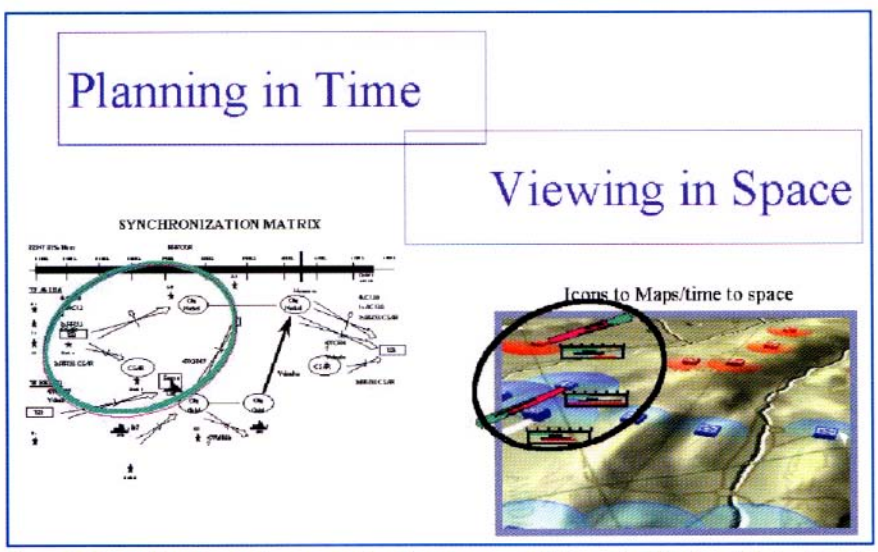
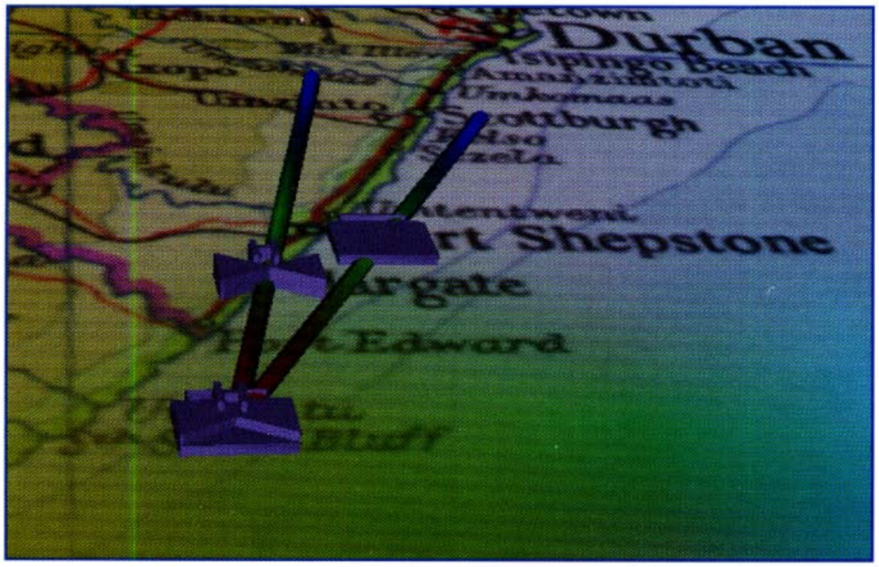


Figure 2. The problem is creating constraints in one conceptual dimension (time) and viewing them in another (space) while providing context for the view.



Figures 3a: Screen capture of a prototype animated map overlay of temporal constraints within plan actions.

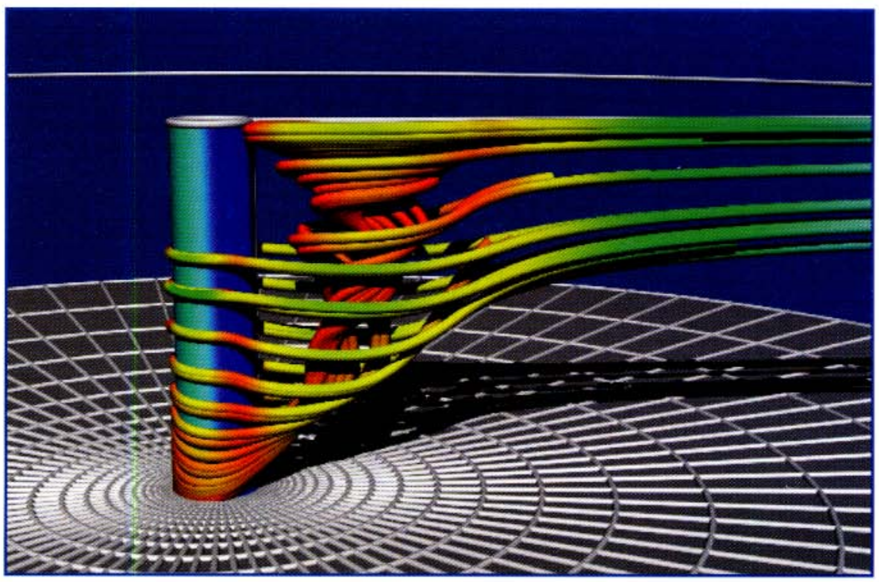


Figure 3b: The same graphical primitives as they are used to visualize fluid dynamics and turbulence about a post.

guage. Our approach uses a layered architecture, with each layer adding increased abstraction while moving closer to the application domain. The integration with Tachyon was easily accomplished using Java wrappers because Tachyon is available as a Java application.

OpenGL is a de facto standard for high-precision computer graphics. OpenGL runs on both Unix and Wintel platforms. Virtually all graphics accelerator manufacturers support OpenGL.

Open Inventor is an object-oriented toolkit for scene database manipulation. It is built on top of OpenGL and is window- and operating system-independent. Open Inventor also contains user interface interaction components and specifies a standard for data interchange. Virtual Reality Modeling Language (VRML) was derived from Open Inventor. Open Inventor uses a scene graph to represent hierarchical geometric models and visual attributes. The underlying architecture is a directed acyclic graph.

The primitives of visualization are the algorithms that transform data into another form. Vtk visualization primitives (each represented as an object in vtk) include:

- **Isosurfaces.** Isosurfaces are triangulated surfaces that represent a constant scalar value. These are the 3D equivalents of contour lines. Isosurfaces can be used to represent radar coverage or surfaces of uncertainty.
- **Glyphs.** Glyphs are 3D models drawn at points in 3D space. Glyph size can be determined by a scalar quantity and oriented by a vector quantity. Logistics, troop strengths, and the like could be represented with glyphs.
- **Streamlines.** Streamlines are 3D lines that follow flow through a 3D volume. Logistics supply lines and atmospheric conditions could be represented.
- **Clipping.** Often just a portion of a representation is required in a visualization. Vtk has a general clipping object that can clip any geometry with any implicit function. A clipper could poke a hole through cloud cover to reveal the underlying logistical plan. Clipping is one critical technique required to implement the information lens (see Figure 1).
- **Probing.** Color is a useful tool for representing scalar quantities. To show one set of scalar quantities on a related surface, we can resample the scalar data set at the vertices of the surface. In vtk, we have isolated this operation into a probe object. A 2D collection of way points for a mission could be placed on a digital elevation map by probing the map with the way point vertices. The vertices could then be deformed according to the scalar elevation.

Unlike a COTS system, vtk is still an evolving toolkit. It has strong architectural and programming structure that can readily adapt to new visualization domains. Tachyon and vtk both employ hierarchical data abstractions and a strong, object-oriented program-

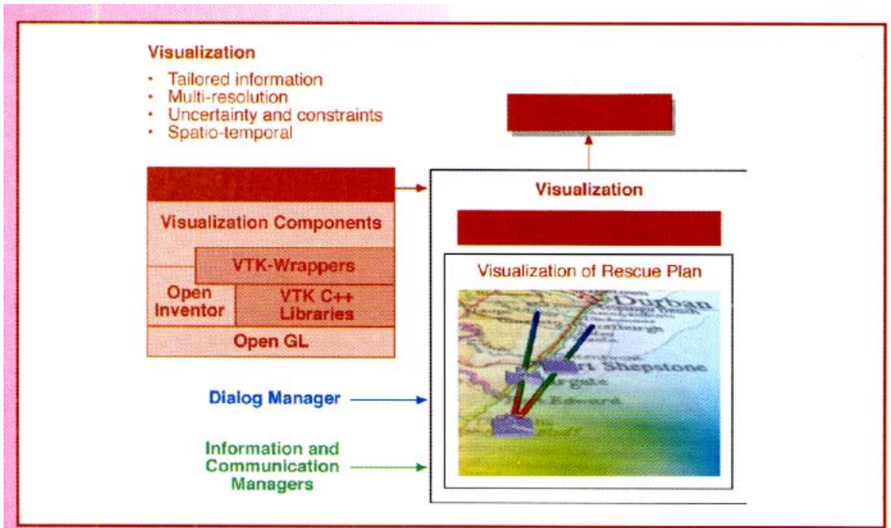


Figure 4. Vtk architecture and the integrated application architecture. Tachyon subsumes the roles of dialog, information, and communication manager.

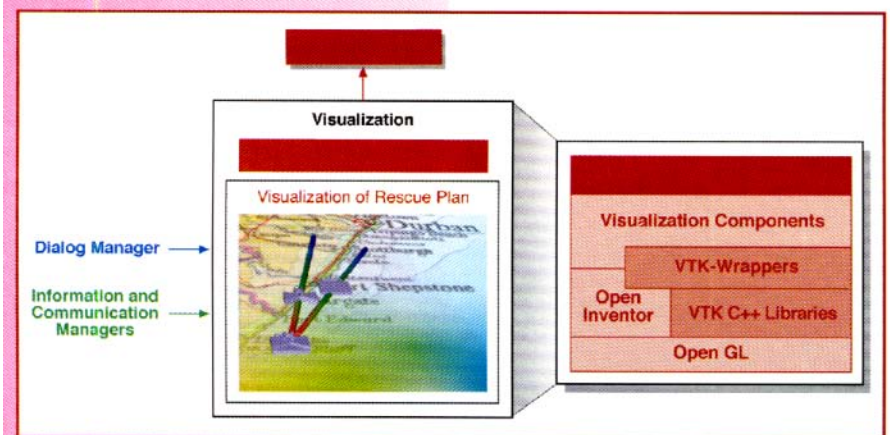


Figure 5. The Tachyon and vtk integration, within vtk's architecture and using Tachyon data, easily provides displays such as multiresolution.

ming structure [see also Rumbaugh et al. 1991] that easily enable new visualizations such as the multiresolution attribute shown in Figure 5.

Tachyon

Tachyon is a constraint-based model for representing and reasoning about qualitative and quantitative aspects of time. It allows substantial expressiveness, provides fast computation over convex intervals, and serves as a tractable testbed for topology-driven techniques for handling calculations over non-convex intervals. Tachyon's implementation features a graphical interface using X-Windows and InterViews. Although Tachyon is a wonderful developer's interface, many users found the graphical interface too technical or scientific-oriented. Tachyon's latest port to Java includes several views and the ability to embed these views into other documents, such as shown in Figure 7. We anticipate that this model and its implementation will find applicability in several areas, including scheduling, project planning, feasibility analysis, and spatio-temporal databases and visualization. [See, for example, Bienkowski and Hoebel 1998].

Visualization of Spatio-Temporal Constraints

The demonstration consists of a plan, represented as temporal constraints on events; spatial locations for the events; and models of the mobility of the military units involved in the events. The constraints show the relations between events, such as *before*, *after*, and *disjoint* [see Allen 1981, Allen 1983]. These constraints are seen as arcs in Tachyon's network view. Unary time constraints may also be represented, such as the minimum or maximum duration of an event or a hard finish time, such as *noon on Day 4*.[3]

The prototype application provides a palette onto which is drawn a dynamic presentation of the constraints of a plan. Vtk is used to draw an animation of the plane. The simplifying assumptions of the prototype are that space is fixed and mobility is uniform. Also, we use scanned image maps rather than actual 3D terrain elevation models. An implementation using 3D terrain data is straightforward and is currently being developed. This will allow for more complex models of mobility rather than the simple one currently used.

The visualization of spatio-temporal constraints consists of a continuously updated display (an animation) of the constraint space in which an event might exist. This is not a simulation because there is no simulation of a causal model. It is not execution of a schedule because no times have been chosen from a set of possible times.

For example, a task force can move inland only after a helicopter has landed it from an offshore transport. Because of the uncertainty of the starting time of the operation, the travel time of the helicopter, and the delay in moving inland, the symbol represent-

[3] For further information on Tachyon's model, the interested reader is directed to [Arthur and Stillman 1992, Arthur and Stillman 1993, Tachyon 1994] and for temporal aspects of planning to [Bienkowski and Hoebel 1998].

ing the task force may have multiple display instantiations at all these points at any one time. That is, the plan constraints in place at noon on Day 4 might allow for a particular task force's being on the transport, en route, or moving inland. We display all of these simultaneously and in a dynamic and changing environment. We are able to show the space of all the possibilities, the entire constraint space, continuously. All temporal data come from Tachyon and are used to drive the dynamic visualization. Tachyon's integration with the vtk data pipeline is live and complete. That is, Tachyon may be called from the spatio-temporal visualization display, using a *picking* or selection capability, to directly interact with Tachyon's data and constraint dialogue interface. This demonstrates the ability of vtk to show a live display of constraints as expressed by a user or any other application for that matter.

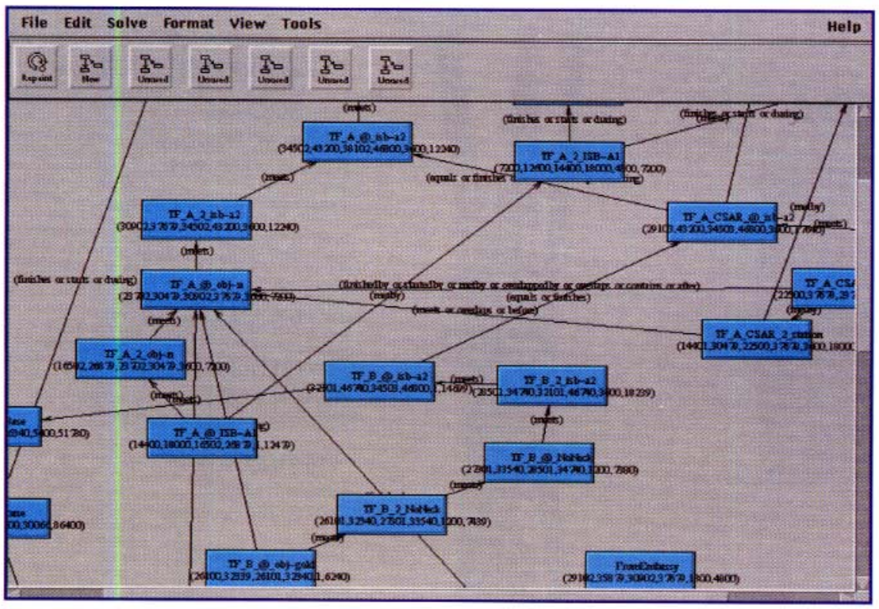


Figure 6. Tachyon user interface, network view.

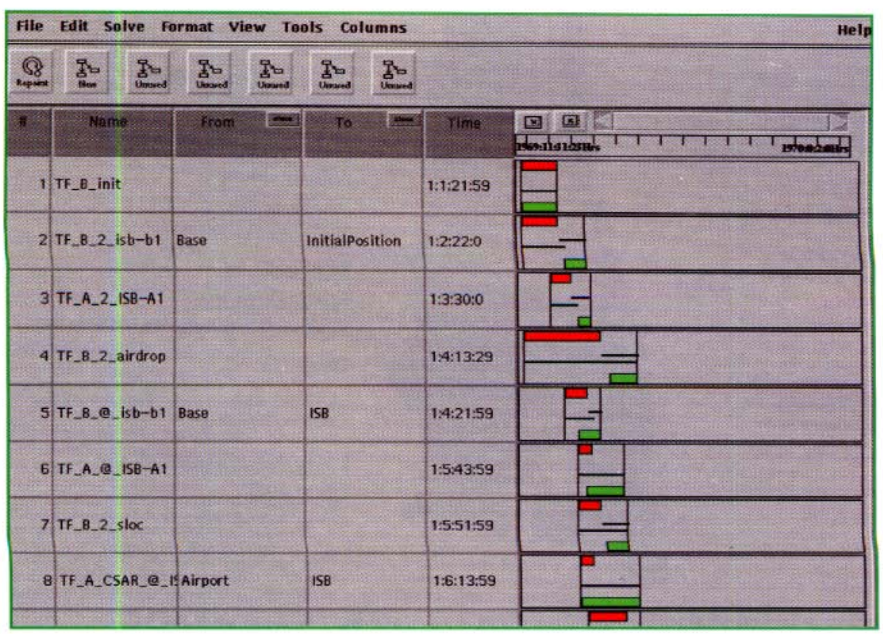


Figure 7. General user dissatisfaction with a technical developer/researcher interface and a desire to produce a more general and transparent view of temporal constraints led to such early prototype visualizations as this TimeLine view, embedded in a formatted text document.

Further information on vtk may be obtained from *http://www.kitware.com*. A full video clip of the demonstration will be available from the *SIGART Bulletin* Web site at *http://sigart.acm.org* after publication.

References and Further Reading

Allen, J.F. 1981. An Interval-Based Representation of Temporal Knowledge. In *Proceedings of IJCAI-7*, International Joint Conference on Artificial Intelligence, Karlsruhe, Germany, August 1983. pp. 741–747.

Allen, J.F. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26:11, pp. 832–843.

Arthur, R., and J. Stillman. 1992. Tachyon: A Model and Environment for Temporal Reasoning. General Electric Corporate Research and Development Technical Report.

Arthur, R., A. Deitsch, and J. Stillman. 1993. Tachyon: A Constraint-Based Temporal Reasoning Model and Its Implementation. *SIGART Bulletin* 4:3 (July).

Bienkowski, M., and L. Hoebel. 1998. Integrating AI Applications in a Military Planner. In *Proceedings of the National Conference on Artificial Intelligence* (July).

Lorensen, W.E., and H.E. Cline. 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics* 21:3 (July), pp. 163–169.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. 1991. *Object-Oriented Modeling and Design*. Prentice Hall, Upper Saddle River, NJ.

Schroeder, W., K. Martin, and W. Lorensen. 1996. The Design and Implementation of an Object-Oriented Toolkit for 3D Graphics and Visualization. In *Proceedings of Visualization'96*. IEEE Press.

Schroeder, W., K. Martin, and W. Lorensen. 1997. *The Visualization Toolkit: An Object-Oriented Approach to 3D Computer Graphics*. 2nd ed. Prentice Hall, Upper Saddle River, NJ.

Tachyon Reference Manual. 1994. General Electric Corporate Research and Development Technical Report. ■