

Collaborative climate 3D viewer

M. Jern & A. Nilsson

ITN, Campus Norrköping, Linköping University, Norrköping, Sweden.

Abstract

Collaborative Geovisualization refers to development of scientific understanding that is mediated by using geovisualization tools in a collaborative context. A “Collaboratory” is a visual computing environment allowing project teams to collaborate and share data and insight while distributed over a network, using intuitive visual navigation techniques “Visual Conference Calls”. The innovation can be summarised as the integration of network technology used in collaborative games with 3D data navigation techniques and based on a low-level, fine-grained “atomic” component infrastructure, implemented and validated in the context of geospatial knowledge construction. We demonstrate how a Collaboratory fuse 2D and 3D data visualization, visual user interface, GIS and collaborative computing into an instinctive interactive weather and climate data analysis experience in which multiple users can participate in an unconstrained process of individual and group discovery.

In this paper we focus on geovisualization tools describing a conceptual overview and a Collaboratory prototype implementation of new methods of presentation and dissemination of terrain, precipitation and flood warning data. A Web based application component for 2D and 3D visualization of climate, DTED, satellite and GIS data that supports peer-to-peer collaborative sessions on PC desktops and laptops has been developed. The paper proposes how collaborative technology, “Collaborative Visual Data Navigation” with web-enabled application sharing, can be used routinely. This project is a blueprint to inform educators, researchers, and policy makers on the effective and sustainable use of interpersonal, collaborative media in science education. We give particular attention to the ways in which dynamic visual representations can be used in these contexts to facilitate shared understanding. The prototype collaborative viewer allows multiple users to view and manipulate the changing climatic data simultaneously and thus to share knowledge.

1 Introduction

"Collaborative Geovisualization" refers to development of scientific understanding, which is mediated by scientific visualization tools in a collaborative context. Our project seeks to understand how, for example, 3D geovisualization could take broad advantage of these capabilities, providing motivating experiences for students and teachers with contemporary visualization tools and topics. Rapid advances in electronic communication technologies that make collaboration at a distance both practical and expected will exacerbate the single-user limitations of existing tools and approaches. Particular emphasis is given in this paper to the role of visual display as a mediator between individuals who apply different perspectives to a problem and the role of visualization methods in providing both participants and system designers with important feedback about the process of collaboration. This paper proposes how one such technology, "Collaborative Visual Data Navigation" with web-enabled application sharing, can be used routinely. After providing a conceptual overview and brief background, we describe the application component of a collaborative 3D GeoVizViewer prototype.

One special goal for our collaborative research is to design an effective and flexible system that can support group work on environmental science research mediated through dynamic visualization displays. We have addressed this goal using a four-step human-centered system design process, modeled on that proposed by Gabbard et al. (1999)[1] for development and evaluation of virtual environments. The steps they delineate are: (1) user task analysis, (2) expert guidelines based evaluation, (3) formative user-centered evaluation, (4) summative comparative evaluation. This multistage evaluation process will help refine the system and its components and reveals usability concerns in the system design process, so that such problems can be eliminated prior to resource intensive, formal user testing.

Our project provides a framework for real time collaborative data visualisation and discovery among geographically distributed remote users, linking people and their desktop in a worldwide "Virtual Data Environment". We define, develop and validate a conceptual distributed computing infrastructure required to support the development of real-time 2D and 3D visual data navigation "Laboratories" based on an "atomic" (low-level) component architecture. The atomic component technology and network infrastructure are based on available Microsoft technology COM, .NET and DirectX's network abstraction and distributed object system. The distributed architecture is based on "Application Component Sharing" providing real-time data interactivity, reducing the load on the network and with zero administration client deployment. Currently, "peer-to-peer" laboratories are supported. Figure 2 shows the overall architecture based on the Viewer, Communication Object and the Collaboratory application component. A client-side component "Collaborative Viewer", responsible for the graphics rendering and collaborative sessions, is developed and distributed free. This free Viewer can be compared with plug-ins such as the Adobe Acrobat Reader or VRML viewers.

2 GeoVizViewer – a Web-based application component

The implementation of dynamic imaging techniques requires new structures for interactivity, data display and dissemination. The radical nature of these new techniques also demands that these structures encompass an inherent collaborative and educational function in order to optimise the understanding of the findings by all members of the multidisciplinary team involved. Current geovisualization data management systems fail to provide the flexibility to combine all the required visualisation and algorithmic technologies in a unitary framework. Most available systems are complex, difficult to use and expensive. Our research project has developed a “GeoVizViewer”, that could offer potential solutions or ideas to some of these problems.

GeoVizViewer is a Web-enabled application component allowing both interactive and collaborative geovisualization examination of climate, DTED, satellite and GIS data and integration with geocomputational methods such as spatial statistics, classification and data mining. The use of Web-based atomic architecture offers the ability to “re-use” low-level algorithmic functions to easily produce application specific components for integration into custom user interfaces. Importantly, the architecture also offers full collaborative interaction and the ability to use low-cost platforms. This approach to system architecture could, for the first time provide the infrastructure for affordable collaborative climate and flood forecasting system interpretation facilities. The GeoVizViewer has been developed to address three application scenarios:

- *Analyse and Explore* – Standalone single user version
- *Collaborative analysis* – Network version
- *Presentation and journaling* – An ActiveX/COM component

2.1 Visual User Interface

Of central importance to our architecture is the Visual User Interface (VUI) that enables the user to directly manipulate graphical objects. The user interacts with 2D and 3D objects with an immediate graphical feedback response, without the need for moving to a secondary menu window. Example of supported manipulation features: 3D orthoplan, dynamic threshold, crop, filter and downsize data, linked 2D and 3D views to support data correlations, drill down through multidimensional data sets, pan, zoom or rotate and color, opacity, visibility. The 3D viewport contains the overview terrain landscape represented as a 3D surface with draped satellite images. The 3D graphics is immediately updated creating the effect of an animation. This 3D viewport provides an overview of the landscape. The area to be investigated is selected by the moving an orthoplan with handlers (figure 1). Data for the selected area is shown in the linked 2D viewport.

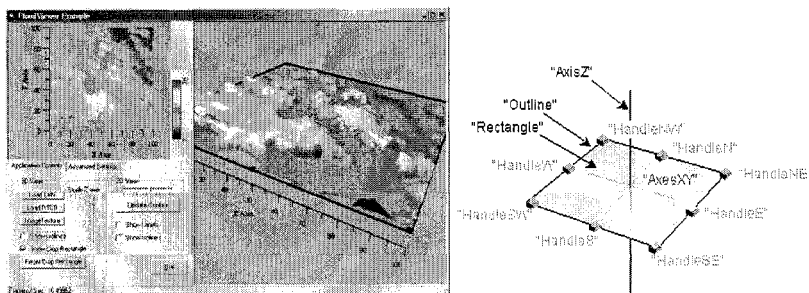


Figure 1: GeoVizViewer – a collaboratory for interacting and visualizing precipitation and flood forecasting data and an Interactor component “OrthoPlaneManipulator”

The OrthoPlaneManipulator atomic component (figure 1) displays the orthoplane as a quadrilateral that is normal to one of the main coordinate axes. The user interacts directly with the geometry of the orthoplane and can click and drag it to update the return value within a pre-defined range. Using this component is similar to using an industry standard slider or similar control to return a value.

3 GeoVizViewer based on a collaborative game architecture

The fundamental challenge when writing a networked, collaborative multi-user application is how to propagate state information to all the desktop computers in the session. The state of the session defines what users see on their desktops and what interactive actions they can perform. The session state is responsible for communicating the data navigation process and environment among the participant users. For example, the selection of multivariate data to be analysed, picking a special data object for additional metadata, correlation between data objects in 2D and 3D space and 3D viewing attributes including rotation, zooming and panning etc. Each user also has a state indicating the user's current properties. Actions performed by users of a data navigation application change the state of the session. When this happens the change will be propagated to the other desktops in the session through messages.

GeoVizViewer is not the only Web-based collaborative application. There are multiple ongoing or completed projects in the area of Web-based distributed systems. The computer graphics research community has devoted many resources to collaborative visualization. Research has focused on Java, VRML and the DIS protocol (IEEE Distributed Interactive Simulation) [2], [3], [4], [5].

The GeoVizViewer extends the Web-based collaborative paradigm to the domain of interactive data navigation in an integrated 2D and 3D worlds and VUI technology supporting large climate, flood forecasting, terrain and satellite image data sets. The project has learned from other research projects that have been dedicated to collaborative games [6]. The network service provider

architecture in the GeoVizViewer application component is based on Microsoft's DirectX API. The GeoVizViewer integrates component-based 2D and 3D geovisualization tools and VUI methods with connectivity software based on emerging industry-standard network protocol for collaborative games. The GeoVizViewer supports most generalized communication capabilities shielding users from the underlying complexities of diverse connectivity implementations, freeing them to concentrate on the real-time navigation scenario. The integration of a network abstraction system, "Application Sharing" and a "Data Navigation Protocol" provide the foundation for real-time collaborative geovisualisation.

Our implementation of a collaboratory provides a layer that largely isolates the user from the problems of an underlying network. With a multi-user application session, each user's VUI is synchronized with that of the other user(s) in the session [7]. Continual stream of messages flow to and from each user. For example, every time a user rotates a 3D object or changes an attribute, a message is sent to update that user's position on the other application client in the session. Our GeoVizViewer collaboratory supports efficient and flexible messaging between all the computers in a session.

3.1 Overall architecture

From the system point of view, the system has a symmetric architecture. To initiate a collaborative session, one of the sites starts up its GeoVizViewer collaboratory. Other participants then start up their own collaboratory to connect to the first participant that is acting as a "server". Each collaboratory application is identical to the others so that any of the applications can run as the "server" when the first participant decides to leave the session. Communication among participants is done through the Communication Object. This kind of "application component sharing" will provide an optimised networked environment with no bottlenecks.

The service provider architecture of the Communication Object insulates the collaboratory from the underlying network abstract layer they are running on. The Communication Object provides all tools needed to support a networked, collaborative data navigation session by providing the services to establish connection between desktops, create and joining sessions, manage single users or group of users, send messages between users, and automatically propagate the state among the computers.

3.2 Viewer Component

The developed collaboratories will be sharing an "engine" or the Viewer plugin component responsible for visualization, interaction, networking and rendering (figure 2). The Viewer component connects directly into the window layout system of the parent application, and takes care of managing all of the visualization inside the view including taking full advantage of and manage the complex interactions with high-performance graphics layer in OpenGL. The Viewer will not only be responsible for drawing and updating the visualization

but also for carrying out all mouse interactions and management of the viewing hierarchy.

Peer-To-Peer Mode

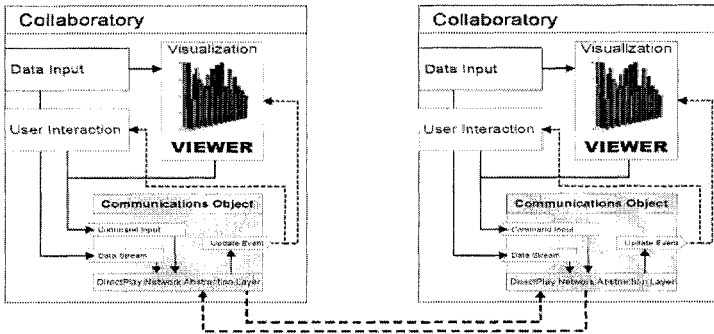


Figure 2: GeoVizViewer collaborative architecture in “Peer-to-Peer” sessions with the central plug in Viewer component, Communications Object and the application component “Collaboratory” (GeoVizViewer).

The GeoVizViewer architecture allows lightweight applications and controls to be deployed across the Web. Downloading the Viewer (6Mb) to the client machine enables this. The GeoVizViewer has a small footprint (200Kb) If the Viewer is not present, it will be automatically installed the first time the application is run. The Viewer plug-in is responsible for:

- Rendering of the output of the components in the Scene Tree to create the visualization
- Management of a system of attributes that control features of *Viewer*, including the color and shading of surfaces, projected images, line widths, styles and so on
- User interaction with the data in the view, including transformation, scaling, and picking
- Interface to the Communication Object responsible for the collaborative session management

The Viewer will be distributed as “freeware” to allow free distribution of collaborative GeoViz applications on the network

4 GeoVizViewer collaborative methods

The GeoVizViewer implements the DirectX classes to create a DirectX object that can start a DirectPlay peer-to-peer session with predefined parameters. The application must create a DirectPlay “player” (the API is originally meant for game play!) to be able to send and receive messages. All communication messages take place between different DirectPlay players and not between

different computers. We call the players “members”, who are identified by a unique ID-number assigned when the user connects.

When connecting, the members first have to select through which service they want to connect. The application enumerates the systems available service providers, for example IPX LAN, TCP / IP, modems or serial cables. When the choice is made, the selected service provider is initialized. When the service provider is selected and successfully initialized the user can choose between joining a session or create a new session and wait for participants to join in. If the user decides to host a session, an instance of a DirectPlay session object, which is really a direct communication channel between the involved computers, must be created. The user that starts a session is called the host and is the only one that can change session parameters, for example max amount of participants. The GeoVizViewer allows DirectPlay host migration. If the host decides to end the session, the other user overtakes the hosting of the session and other users can join it.

4.1 Active and Passive members

When contact has been established and the session is running, one of the players is the host and the other is what we call a “*passive member*”. The host is the only member that can interact with the graphics and change the parameters of, for example, the 3D surface and is therefore called the “*active member*”. Although the underlying technology is peer-to-peer, we have used client-server philosophy when building the application. All interaction on the host side is briefly stored and sent to the passive member. The messages are received and the appropriate parameters are updated and the application is invalidated. The passive member is actually just a spectator watching the interaction by the active member.

The host, which is the active member of the session as default, has the ability to decide which one of the members that will work as the session’s active member. When the host approves a passive member to be active, the host becomes a passive member but, unlike other passive members, has the ability to retake the active membership. The reason why we implemented the application in such way is to make sure that only one user modifies the interactive tools and data. The purpose of the application would be lost if multiple users affected the data and view at the same time. The GeoVizViewer includes a simple chat so that communication between the users is possible. In further work, an implementation of for example DirectPlay Voice would be suitable for the application.

4.2 Messages

When a message package arrives to a user it’s stored in a message receive queue. The GeoVizViewer extracts the application messages from the queue by an identifier that determines the type of message (Figure 3). A number of different message types are handled. Since the main objects of the GeoVizViewer is the 3D surface and the interactive orthoplane cropping interactor, some of the main

message types contains the values of the parameters connected the 3D surface object and the interactor respectively. The main GeoVizViewer messages are:

3D surface message: The message contains data about isovalue, surface opacity, surface color and surface object visibility.

Orthoplane message: The message contains data about slice opacity, slice color map, plane location and plane visibility

Matrix message: The message contains information about the transformation matrix. Every time the objects are affected by translation, rotation or zooming this message is sent to the other user.

Annotation message: The message contains annotation data. It is used in three different ways: 1) When the session have started and the users are exchanging annotation data. 2) When new annotations are set 3) When an annotation is modified

The File information message: The message contains information about the data sets that is loaded by the host. If the file dimensions doesn't agree with the data file that the other member use the session cannot continue. Different data files in the same session could cause application errors.

There are also other types of messages that deal with other parameters such as background color, viewport location and chat messages.

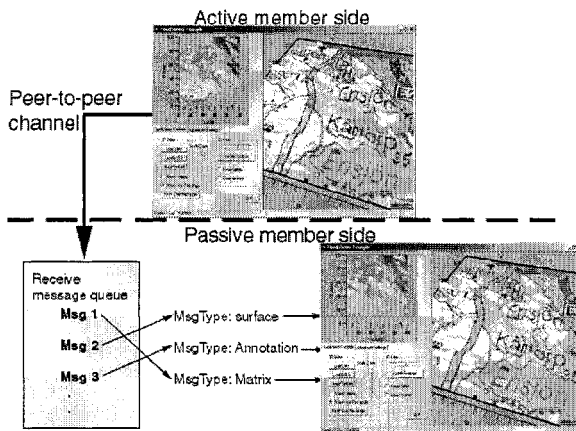


Figure 3: Illustration of the message flow between the active member and the passive member

The messages are built the same, but contain different data. A message consists of a buffer that holds the data. The first thing added to the buffer is the message type. This is the first thing that is checked when the message is received to ensure that the right action will be performed. The content is sent (and received) as data or strings. Strings cannot be sent as data. Therefore the message type identifier is very important to ensure that it is known which kind of data the message contains. In many cases more than one data value is sent. An offset variable is inserted after each data as a separator between the different data types.

4.3 Information Protection

The original parameters, like opacity, surface value etc, cannot be restored after a session except for the rotation matrix. The annotations are also handled with extra care. During a session, the annotations from the user and from the other member are separated to avoid name conflicts. Another important reason is to make sure that no one but the user that created the annotation can remove or modify it. With the user's own annotation in one annotation component and the other member's annotations in another, the application can control that only the "member" annotations can be edited by the participating member. Thus, a user can only edit his/her own annotations while a session is running. Annotations from both sides are of course visible, displayed and updated at both sides.

4.4 Annotation tool

An annotation tool has been developed to provide the ability to draw or mark out regions of interest on 2D or 3D objects currently being viewed in the collaborative environment. The annotation tool is needed if the Visual Conference system is to be of any practical use. The annotator allows you to place colored text, arrows or polyline anywhere in space, or, to attach that text etc to a particular object. Thus, if the 3D surface is later moved or rotated, the annotation will remain in its correct position relative to the 3D surface. It is important that only one user at a time can access the annotator, however, otherwise the behavior of the GeoVizViewer is undetermined. The underlying network system will solve this contention problem.

5 Conclusion

We are not trying to invent yet another collaborative system and network protocol, but have instead built on existing network software in DirectX and demonstrated how this system originally developed for collaborative games can also be used for geovisualization. Various research tasks will contribute to the following innovative results in several areas including:

- Implement, test and validate a distributed network architecture based on "application component sharing" and communication software for multiplayer games to be applied in geovisualization
- Implement and validate GeoVizViewer in a global environment with different level of expertise and culture
- Data interaction styles and their implications - including data correlation, data filters, classification, clustering and fast data reduction techniques
- Consistency of cognition models across information appliances
- Usability issues and measuring the effectiveness of symbiosis - the demonstrators will revolve around a project team being able to collaborate over a whole archive of project data

The key issues that we are trying to address, are adoption patterns, impacts on performance and process, problems created by the technology, and innovative solutions. We try to better understand how teams adopt and use collaboratories in their work setting. Do teams with different expertise experience different patterns? What problems do teams experience while collaborating through GeoVizViewer, and how do they circumvent them? Can the user interface be improved? Based on our work so far, we have identified the following research challenges that must be met to make effective collaborative visualization possible, to:

- Develop a theoretical understanding of the cognitive and social aspects of both local and remote collaboration mediated through display objects in a medical and other planned application context
- Assess the advantages and disadvantages of extending methods of interactive 3D imaging and geovisualization methods (developed for single users) to collaborative settings versus designing new methods to meet unique characteristics of group work
- Implement client-server solutions for multi-user applications, e.g. for educational applications.

Acknowledgements

Research was supported by two EC funded projects MUSIC (VK1-CT-2000-00058) and SMARTDOC (IST-2001-28137). See more details at the web sites [8], [9].

References

- [1] Gabbard, J. L., D. Hix, and J. E. I. Swan., 1999. User-centered design and evaluation of virtual environments. *IEEE ComputerGraphics and Applications* Nov/Dec:51-59.
- [2] Wood, J., Wright, H. and Brodlie, K., 1997. Collaborative Visualization, Proc., IEEE information Visualization '97. IEEE Computer Society, Pheoniz, Oct. 19-24, 1997, pp. 253-259.
- [3] 1999. Learning and building together in an immersive virtual world. *Presence*, 8(3): 247-263.
- [4] Hindmarsh, J., Fraser, M., Heath, C., Benford, S. and Greenhalgh, C., 2000. Object-focused interaction in collaborative virtual environments. *ACM Transactions on Computer-Human Interaction*, 3(4): 477-509.
- [5] Padula, M. and Rinaldi, G.R., 1999. Mission critical web applications: A seismological case. *Interactions*, July-August: 52-66.
- [6] Di Benedetto, DirectPlay 8 Overview, Microsoft Corporation, May, msdn.microsoft.com
- [7] Jem, M., 1998. "Thin" vs. "fat" visualization client, Proc., Computer Graphics International. 772 -788.
- [8] SMARTDOC web site: <http://servus.itn.liu/smartdoc/>
- [9] MUSIC web site: <http://www.geomin.unibo.it/orgv/hydro/music/index.htm>