

# Virtual reality and dynamic statistical graphics: A bidirectional link in a heterogeneous, distributed computing environment

Martin Schneider

Philipps-Universität Marburg

Dept. of Mathematics and Computer Science  
Hans-Meerwein-Str., 35032 Marburg, Germany

Jürgen Symanzik

Utah State University

Dept. of Mathematics and Statistics  
Logan, Utah 84322-3900, USA

Christoph Stamm

ETH Zurich

Institute of Theoretical Computer Science  
ETH Zentrum, 8092 Zurich, Switzerland

Peter Widmayer

ETH Zurich

Institute of Theoretical Computer Science  
ETH Zentrum, 8092 Zurich, Switzerland

**Abstract** *We present the implementation of a bidirectional link between RA<sub>3</sub>DIO and XGobi, based on the middleware standard DCE. RA<sub>3</sub>DIO is a virtual reality framework for the design and management of mobile phone networks and the optimization of antenna positions. XGobi is a software package particularly useful for the analysis of multivariate statistical data. RA<sub>3</sub>DIO only runs under a PC Windows environment and XGobi under Unix. We describe the main functionality of the link and related topics such as concurrency and linked brushing, and we discuss the underlying internal communication structures. Finally, we recommend some software design rules with respect to our experiences with the presented link.*

**Keywords:** Distributed computing; Interoperability; Multivariate statistical data; Remote procedure call (RPC); Visual data mining; Wave propagation.

## 1 Introduction

For Geographic Information Systems (GIS), an exploration of a virtual world is a powerful metaphor for interacting with the system. A part of such an exploration mechanism is readily available: Visual realism and real-time navigation are provided by typical virtual reality systems. Database functionality, including efficient access to huge amounts of geometric and texture data on disk, can be added with sophisticated coupling mechanisms [1]. With

such a coupling, our application of interest can be fully supported: Mobile phone antenna positions and the induced signal strength distribution on a terrain can be visualized and explored in a virtual trip in real-time, while the antenna and terrain data can be maintained in the database [2]. It should be obvious that a system with this functionality is useful to a large part of the population today, since it helps to identify mobile phone antenna places that keep the potential adverse effects on the people at a minimum.

The information that such a system provides, however, is in many cases incomplete without proper statistical evaluations. For the influence of a mobile phone antenna, for instance, it is important to know its activity pattern: When is the antenna in use, and to what degree? What does this imply on the actual signal strength distribution over time and space, as compared to the maximum signal strength? These questions and many others might arise in the antenna placement decision process, and it is certainly wise to have a powerful off-the-shelf statistics package available to answer them. Such a package must interact properly with the virtual reality/database system. This is easier said than done: The user might want to have several windows on the screen, one with the virtual reality, another with a histogram of antenna usage, and as he/she moves through the scene, he/she might want the histogram to dynamically change

according to what he/she sees. If there is an interesting pattern shown in the histogram, the user might want to have the system highlight it in the virtual reality window, and vice versa. This latter functionality is called “linked brushing”, since the selection (by brushing) of an item in one window also selects the same item in all other windows. It is easy to imagine that with such a system, far more interesting conclusions can be drawn than with a system that lacks any of these components.

Unfortunately, it is by no means obvious how to connect the components in the best possible way. Certainly, it is neither doable nor desirable to change the source code of two systems in order to make them communicate. In this paper, we report on the issues involved in a system interaction mechanism that we designed and implemented for the application above: A mobile phone antenna placement and management system, RA<sub>3</sub>DIO, interacts with a statistical graphics package, XGobi. The acronym RA<sub>3</sub>DIO stands for “Radio Antenna placement with 3-Dimensional Interactive Optimization”. RA<sub>3</sub>DIO is based on the research prototype WorldView [3]. It has been implemented to visualize and explore virtual terrains and terrain related themes, especially electro-magnetic wave propagation of transmitters in rural and suburban areas. Microsoft Windows has been chosen as platform for RA<sub>3</sub>DIO because of its wide distribution and its huge number of standard components. Therefore, it runs both on fast graphic workstations and on small Notebook computers.

XGobi [4] can be best described as a high-interaction statistical graphics package. These packages have been developed within the last 15 years to enhance the graphic facilities developed for exploratory data analysis since the early 1960’s. XGobi is mainly available for Unix platforms.

Our link connects RA<sub>3</sub>DIO and XGobi with Remote Procedure Calls (RPCs). There are several possibilities to link or couple different programs. We call a coupling loose if the communication is realized by reading and writing shared data files. In contrast, a close coupling is based on built-in software components, which can be invoked by another program [5]. RPC is a typical method of close coupling. For a fast working link that has to send information in real-time, close coupling is

the method of choice.

RPCs are often integrated among other services in a middleware. We considered several middleware standards like CORBA [6], DCE [7], DCOM [8], and Java RMI [9], as described in Section 4. We decided to use DCE (Distributed Computing Environment), because it is efficient and independent of the operating system and network.

The remaining paper is structured as follows. The next two sections introduce the programs we have linked. Section 4 is about DCE, the underlying communication method used for our link. Section 5 ensures that the reader is able to understand the ideas that are underlying the link between RA<sub>3</sub>DIO and XGobi. It also provides insights how to construct similar linked software environments and looks at possible applications of our link. Section 6 concludes this paper and Section 7 gives an overview on possible future work.

## 2 XGobi

XGobi is a dynamic statistical graphics program that can be used for visual data mining. Some examples where XGobi has been successfully used are the display and clustering of shopping-frequency data [10] and the visualization of experimental measurements of laser performance and high-dimensional geometric objects [11]. XGobi can be used in applications as different as the classification of archaeological sites and for human motion data.

One of the main features provided by XGobi is the grand tour [12]. In addition to the standard grand tour, XGobi also supports the projection pursuit guided tour, a combination of two complementary methods into an interactive and dynamic framework. Another main feature is “linked brushing” of multiple XGobi windows that are displaying different projections of the same data. “Linked brushing” is helpful in the exploration of clusters in high dimensions. The term “linked brushing” refers to the concept of marking points in one window using different symbols and colors and automatically updating all “linked” views.

### 3 RA<sub>3</sub>DIO

RA<sub>3</sub>DIO is a virtual reality framework for electromagnetic wave propagation based on a terrain explorer for very large topographic scenes. The visualized terrain is a combination of adaptively triangulated digital elevation data and texture (e.g., aerial ortho-photos, topographic pixel maps). The texture helps the user to better recognize terrain parts. The concepts that are required for the terrain explorer to handle huge amounts of terrain and texture data in a virtual reality manner (dynamic scene management, level of detail management) have been extensively studied in the ViRGIS project [1]. RA<sub>3</sub>DIO makes use of the same concepts.

A link between RA<sub>3</sub>DIO and XGobi is very useful and interesting. It offers the possibility to analyze the various kinds of data provided by RA<sub>3</sub>DIO. An actual example are radiation emissions. RA<sub>3</sub>DIO can provide a large amount of data for visual exploration in XGobi, e.g., the terrain data (polyhedral triangulated data) on its own or the spatial data objects handled in RA<sub>3</sub>DIO, such as cities with their parameters (city name, spoken language, number of habitants, city area, etc.) or the antenna data set (position, direction, height above ground, power, antenna type, carrier frequency, etc.). Even combinations of the previously mentioned multivariate, spatially referenced data objects can serve as a source for an exploratory data analysis in XGobi.

### 4 Distributed Computing Environment (DCE)

At the moment, there are several competing middleware standards. One of them is the OSF (Open Software Foundation, now The Open Group) DCE. This is a key technology in three of today's most important areas of computing: security, the World Wide Web, and distributed objects. It is the only suite of integrated services from a vendor-neutral source that enables organizations to develop, use, and maintain distributed programs across heterogeneous networks [13].

We have considered alternative standards to

DCE such as the Common Object Request Broker Architecture (CORBA) [6] from the Object Management Group (OMG), the Distributed Component Object Model (DCOM) [8] from Microsoft, and Java RMI (Remote Method Invocation) [9]. Due to the space limitations, we omit the advantages and drawbacks of the alternative middleware standards.

#### 4.1 Remote Procedure Call (RPC)

DCE provides a client/server model. A service exporter registers services at a server and a service importer can bind to those services. All DCE services are based on the RPC mechanism which is typical for middleware.

Although the connected machines can have different data formats, there is no explicit data conversion necessary in the program's source code. DCE automatically converts the data in the correct way during the marshalling/unmarshalling process.

In DCE, a program can offer a set of RPCs. It has to declare them in an interface specified in the Interface Definition Language (IDL). The IDL compiler generates a client and a server stub, which are linked to the client and to the server, respectively. The stubs perform the "real work", the platform dependant communication, invisible for the programmer. IDL also generates some procedure declarations that let RPCs look like simple local procedure calls. Only the initialization and the finalization contain DCE-specific code.

Each interface has a universal unique identifier (uuid) built from the network address and the system time at the time of the creation. This ID is important for the interface registration at a DCE server machine. After the registration, the server program runs into a status waiting for incoming calls, where it is possible to answer multiple calls at once in multiple threads. A client can connect to the server and bind to an interface at run-time.

### 5 The RA<sub>3</sub>DIO/XGobi Link

Our implementation is based on the ideas of the previously developed link between ArcView, a GIS, and XGobi [14, 15, 16]. This link provides

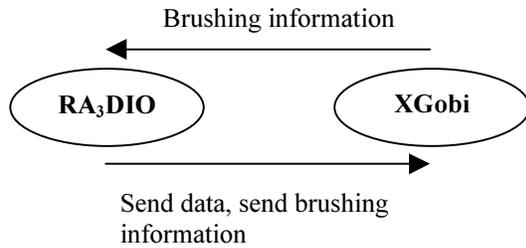


Figure 1: Schematic RPC Communication

several different features to display spatial data in XGobi. One feature, the “basic link”, simply transfers multivariate data from ArcView into XGobi where it is displayed and can be interactively manipulated. For the other features, XGobi performs a pre-computation after the data has been transferred from ArcView. For example, XGobi computes the variogram-cloud plot or spatially lagged scatterplots [17].

The pre-computation in XGobi is a little bit historic because ArcView’s script language Avenue was slow and not very good prepared for complex computations. The pre-computation means for our application that XGobi provides extended possibilities compared to the standard XGobi. All features of XGobi that are available in the ArcView/XGobi link can also be accessed in the RA<sub>3</sub>DIO/XGobi link.

The main interactive function of our new link is the so-called “linked brushing”. This function is characteristic for the bidirectional link: On one side, user actions in RA<sub>3</sub>DIO may change the data and the brushing information in XGobi. On the other side, interactive brushing that occurs in XGobi may result in data changes in RA<sub>3</sub>DIO.

Both, XGobi and RA<sub>3</sub>DIO, are working as DCE client and DCE server at the same time. Therefore, they can communicate in both directions (see Figure 1). XGobi provides 35 remote procedures as a server and calls one remote procedure as a client (update brushing information). We will describe some of the remote procedures in the following sections in more details.

### 5.1 The XGobi Side of the Link

The “basic link” between XGobi and an external program (e.g. RA<sub>3</sub>DIO) consists of nine

remote procedures that are provided by XGobi. First, XGobi has to be started in a remote mode. Then, by calling “RPC\_Init\_Data”, the client transfers the name, the number of data points, and the number of dimensions of the initial data set to XGobi. “RPC\_Send\_Init\_Data” is the procedure to transfer the data to XGobi. With “RPC\_Send\_Init\_Symbols”, XGobi can be informed about the initial color and/or symbols that are used to display the data points. After the “RPC\_Make\_XGobi” call, XGobi opens an X-window and gets ready for user interactions.

The biggest problem for XGobi was the synchronization with the RPCs. XGobi is a single-threaded application. That means, only one thread can access internal data structures at a time. In DCE, RPCs can occur at any time. So, there has to be a synchronization between each RPC thread and the main XGobi thread. This is the classical mutual exclusion problem that can be solved using semaphores.

In our solution, every RPC first runs into a lock statement. This means that the thread stops until the semaphore variable is unlocked by another thread, i.e., by the XGobi main thread. XGobi calls the unlock statement many times a second. Then, a waiting RPC can start with its execution. The XGobi thread becomes suspended at the following lock statement until the RPC finishes its computation and unlocks the semaphore again. This method works in the same way if multiple RPCs (from different clients) are allowed at the same time.

### 5.2 The RA<sub>3</sub>DIO Side of the Link

The RA<sub>3</sub>DIO side of the link is substantially simpler than the XGobi side because RA<sub>3</sub>DIO is a thread-safe multithreaded program. Thus, the thread-based DCE-RPCs cooperate very well with the behavior of RA<sub>3</sub>DIO.

After a short initialization of the link, RA<sub>3</sub>DIO is immediately ready to send the initial data set to XGobi and to open XGobi either on the PC with a X-window emulator or on a Unix workstation. During the initialization of the link, a thread handler must be started. This handler invokes a number of concurrent threads to serve all the incom-

ing remote procedure calls. In combination with XGobi, “Xfer\_Brushinfo” is the only remote procedure used on the RA<sub>3</sub>DIO server side.

Once XGobi has been initialized, the user is able to interact with the multivariate data set sent by RA<sub>3</sub>DIO. While brushing some data points of the data set, XGobi calls “Xfer\_Brushinfo” to inform RA<sub>3</sub>DIO about the updated brushing information. The updated data points are numbered according to the order initially used and exactly these ordinal numbers together with the brushing information (symbol, size and color) are sent to RA<sub>3</sub>DIO. RA<sub>3</sub>DIO maps these numbers to the objects using an array and updates the specified appearance of the objects. Because XGobi only sends the ordinal number of the data point, RA<sub>3</sub>DIO has to store the array for the entire time while communicating with XGobi.

We have already mentioned that RA<sub>3</sub>DIO visualizes only a small part of the whole terrain data at the same time. Thus, it makes sense only to transfer the visible data into XGobi. While exploring the terrain, new terrain patches are loaded and patches that are no longer used are removed. Hence, the data set in XGobi does not anymore represent the visualized data in RA<sub>3</sub>DIO. In this situation, it would be desirable that XGobi can handle dynamically changing data. But it cannot. Therefore, we have to reinitialize the whole data set stored in XGobi and reinitialize the array to map the new ordinal numbers to the new objects.

### 5.3 Interaction between RA<sub>3</sub>DIO and XGobi

Figure 2 shows a simplified chart of the interaction between RA<sub>3</sub>DIO and XGobi. A more detailed description of the RPCs and related issues can be found on our web pages.

Figure 3 shows the link at work. The XGobi view (smaller window on the left) shows the 3D-coordinates of the antennas visible in RA<sub>3</sub>DIO. Points in XGobi are brushed using different symbols and colors and the corresponding spatial locations are marked with the same symbols and colors in RA<sub>3</sub>DIO. In addition, the link between RA<sub>3</sub>DIO and XGobi allows the user to explore the terrain and view one or multiple additional variables re-

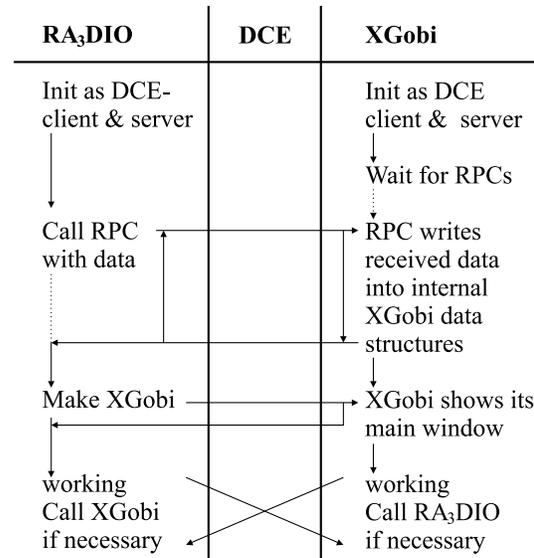


Figure 2: Interaction between RA<sub>3</sub>DIO and XGobi

lated to the spatial location of the antennas in the XGobi window. This extra data can be the power of each antenna or the radiation it emits. XGobi allows to brush points that relate e.g. to antennas with high radiation emissions in its view while through linked brushing, the corresponding spatial locations become highlighted in the RA<sub>3</sub>DIO map view.

## 6 Conclusion

During this project, we studied and solved the problem of connecting two different software packages. We linked the virtual reality framework RA<sub>3</sub>DIO and the dynamical statistical graphics package XGobi. The users of RA<sub>3</sub>DIO can now enjoy to see additional statistical data in XGobi and analyze it with a professional tool.

At the present time, it is not necessary that every software package provides every useful function by itself. Instead, it is more economical to provide an open interface to off-the-shelf software packages so that other programs can use these special features. If the specification of the interface is public, a program can conquer new applications years after the programmer wrote the last line of its source code – even if the source code is not available to others.

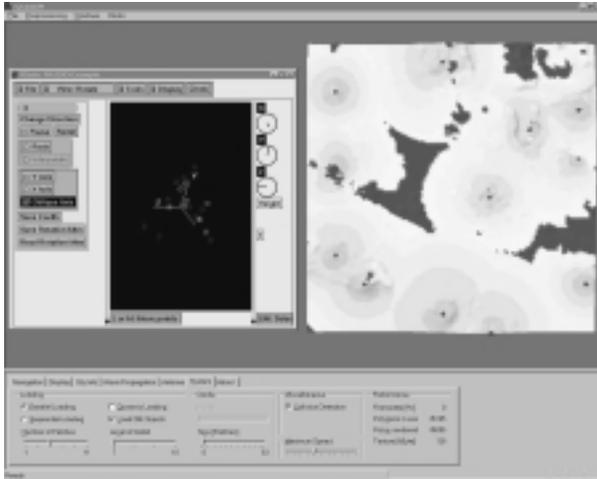


Figure 3: Bidirectional link between RA<sub>3</sub>DIO and XGobi at work

To link the two software packages, RA<sub>3</sub>DIO and XGobi, we used a method of close coupling because the data throughput is a key component in a virtual reality system. Remote Procedure Calls are such a method of close coupling. Among different middleware standards providing RPCs, we chose DCE-RPCs because of their independence of the operating system and network and because of their communication speed.

Our work has shown that it is an advantage to think about middleware already at the stage of the design. Many programmers still build huge monolithic systems that do not provide connections to the world outside. It should be of a greater interest how to design interfaces of open systems [18] and how to connect them to program specific functionality.

Recently, the DCE version of XGobi has been included into the officially distributed XGobi source code. With some additional knowledge that is provided through our web pages, it should be easy to connect additional software systems to XGobi, using DCE-RPCs.

## 7 Future Work

There is a future need to extend XGobi's functionality of the link, because the "send data" protocol of the link was originally not designed for dynamic data updates. This means that in the current imple-

mentation RA<sub>3</sub>DIO has to resend all data of the actual matrix of patches to XGobi and not only the data of the newly loaded patches since the last visible scene update. This results in a superfluous and excessive network load. An interim solution to handle this problem could be a data mirror residing on the same machine as the XGobi client.

Other future work will focus on the CORBA compliant standards that will be provided by the Omega project [19]. So far, there is no common standard for linking statistical software packages and, therefore, every link had other main interests. CORBA can integrate these different viewpoints and should be the future basis for all links from application programs to statistical software packages.

Another direction of our work deals with the portation of XGobi to the PC. The main problem hereby is the RPC part. DCE-RPCs, respectively ONC-RPCs, have to be called from a GNU GCC-compiled XGobi. We hope that we can present our link completely running on only one PC in the near future.

## Acknowledgements

Martin Schneider's research was funded by Prof. Dr. Edward Wegman's NSF Group Infrastructure Grant DMS-9631351 and by the project "Chorochronos" of the European Union. Jürgen Symanzik's work was also funded in part by the same NSF grant. The research was partially carried out while Martin Schneider was visiting the Center for Computational Statistics at George Mason University, USA, and the Institute of Theoretical Computer Science at the ETH Zurich, Switzerland.

## Appendix

The main web addresses for RA<sub>3</sub>DIO and the ArcView/XGobi link are:

[www.ra3dio.ethz.ch](http://www.ra3dio.ethz.ch)

[www.public.iastate.edu/~arcview-xgobi](http://www.public.iastate.edu/~arcview-xgobi)

Additional informations about the RPCs supported by XGobi can be found at:

[www.galaxy.gmu.edu/~maschn/xgobi](http://www.galaxy.gmu.edu/~maschn/xgobi)

## References

- [1] R. Pajarola, T. Ohler, P. Stucki, K. Szabo, and P. Widmayer. The Alps at your Fingertips: Virtual Reality and Geoinformation Systems. In *Proceedings 14th International Conference on Data Engineering, ICDE '98*, pages 550–557. IEEE, 1998.
- [2] S. Eidenbenz, C. Stamm, and P. Widmayer. RA3DIO – Wellenausbreitung in 3D. *Computerworld Schweiz*, 37:26, 1999.
- [3] M. Beck, S. Eidenbenz, C. Stamm, P. Stucki, and P. Widmayer. A Prototype System for Light Propagation in Terrains. In F. E. Wolter and N. M. Patrikalakis, editors, *Proceedings of Computer Graphics International*, pages 103–106. IEEE Computer Society, 1998.
- [4] D. F. Swayne, D. Cook, and A. Buja. XGobi: Interactive Dynamic Graphics in the X Window System. *Journal of Computational and Graphical Statistics*, 7(1):113–130, 1998.
- [5] M. F. Goodchild, R. P. Haining, and S. Wise. Integrating GIS and Spatial Data Analysis: Problems and Possibilities. *International Journal of Geographical Information Systems*, 6(5):407–423, 1992.
- [6] R. Otte, P. Patrick, and M. Roy. *Understanding CORBA: The Common Object Request Broker Architecture*. Prentice Hall, Upper Saddle River, NJ, 1996.
- [7] A. Schill. Distributed Platforms. *Encyclopedia of Microcomputers*, 22:97–116, 1999.
- [8] G. Eddon and H. Eddon. *Inside Distributed COM*. Microsoft Press, Redmond, WA, 1998.
- [9] Sun. Java RMI Tutorial. Internet: <http://java.sun.com/docs/books/tutorial/rmi/>, 2000.
- [10] M. A. Koschat and D. F. Swayne. Interactive Graphical Methods in the Analysis of Customer Panel Data (with Discussion). *Journal of Business and Economic Statistics*, 14(1):113–132, January 1996.
- [11] A. Buja, D. Cook, and D. F. Swayne. Interactive High-Dimensional Data Visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, January 1996.
- [12] D. Asimov. The Grand Tour: A Tool for Viewing Multidimensional Data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143, 1985.
- [13] The Open Group. The Open Group Portal to the World of DCE. Internet: <http://www.opengroup.org/dce/>, 1999.
- [14] D. Cook, J. J. Majure, J. Symanzik, and N. Cressie. Dynamic Graphics in a GIS: Exploring and Analyzing Multivariate Spatial Data Using Linked Software. *Computational Statistics: Special Issue on Computer-aided Analysis of Spatial Data*, 11(4):467–480, 1996.
- [15] D. Cook, J. Symanzik, J. J. Majure, and N. Cressie. Dynamic Graphics in a GIS: More Examples Using Linked Software. *Computers and Geosciences: Special Issue on Exploratory Cartographic Visualization*, 23(4):371–385, 1997. Paper, CD, and <http://www.elsevier.nl>.
- [16] J. Symanzik, J. J. Majure, D. Cook, and I. Megretskaia. Linking ArcView 3.0 and XGobi: Insight Behind the Front End. Technical Report 97–10, Department of Statistics, Iowa State University, Ames, Iowa, 1997.
- [17] J. Symanzik, I. Megretskaia, J. J. Majure, and D. Cook. Implementation Issues of Variogram Cloud Plots and Spatially Lagged Scatterplots in the Linked ArcView 2.1 and XGobi Environment. *Computing Science and Statistics*, 28:369–374, 1997.
- [18] O. Günther. *Environmental Information Systems*. Springer, Berlin, Heidelberg, 1998.
- [19] D. T. Lang. The Omega Project for Statistical Computing. Internet: <http://www.omegahat.org>, 2000.