A Visual Programming Environment for Supporting Scientific Data Analysis *

Yuhong Zhang, Matthew Ward, Nabil Hachem, and Michael Gennert
Computer Science Department
Worcester Polytechnic Institute

Abstract

Our research on a conceptual model for scientific data analysis concludes that traditional computational and environmental support is insufficient. We believe an integrated system that provides users with the ability to manage data, processes, and experiments as well as program data analysis tasks is necessary. In this paper the scientific data analysis process in global change research is analyzed and the development of GaeaVE, a visual environment for a scientific database management system for global change research, is described. The components of the environment consist of a browser, a visual language editor for processes and experiments, and a data viewer.

1 Introduction

Scientific data analysis is the process whereby scientists from various disciplines apply mathematical, logical, and analytical methods to data to generate new information or knowledge in their disciplines [16, 11]. Data analysis is a critical phase in scientific research, and managing experiments and the data produced throughout an experiment life-cycle has become a bottleneck of many experimental studies[13]. Current computer-based support systems for scientific data analysis can be found in two areas: Scientific Databases and Scientific Visualization. Scientific Databases are a relatively new research direction in database management systems development, where the data management issues such as data storage, data access, and data query techniques are studied. Scientific Visualization has a longer history, where many visualization techniques have been developed and organized into various software systems.

One problem with current support systems is that there is no uniform environment for scientists to conduct research without having to learn many different software systems. We believe that an integrated system that provides users both data management ability and data analysis programming ability is necessary. The objective of our research is to use visual techniques in an integrated data analysis and management support system to provide scientists with a complete research environment.

Gaea, a scientific database management system for supporting geographic information analysis and global change research, is a project under development at the Computer Science Department of Worcester Polytechnic Institute [9]. Gaea is designed as an integrated data and analysis management system. It has two components; the Gaea Kernel provides the database management support, while the GaeaVE is a visual environment that provides its users with the ability to program their tasks and to perform management functions over data, operators, and experiments.

A visual environment is a programming environment in which the programming task is made easier for users by using visual representations in the specification and control of computer systems [19]. Several visualization systems provide visual environments for supporting data analysis, including IRIS Explorer [20], Khoros [23], and AVS [22, 1]. Similarities between these systems include:

- Providing both visualization and analysis functionality.
- Providing an extensible data processing and data visualization library.
- Allowing users to build custom applications without writing code in a traditional programming language.
- Using data-flow diagrams as the analysis programming language as well as other visual techniques in the user interface.

In the use of visual representations, the systems mentioned above have many good ideas which are being incorporated into GaeaVE. IRIS Explorer uses process

*This work was partially supported by NSF contract IRI-9116988

menu icons in a data flow diagram to make the specification of a particular process an easy and intuitive task. AVS makes three major display processes into data viewers: the geometry viewer, the image viewer and the graph viewer. In this way, the interface provides users at different levels a different working environment. An application user, who uses only the existing processes, does not have to know the detailed procedure for defining a new task. In AVS, operators are categorized by function, and color is used to indicate input and output data types. The goal of Khoros is to integrate research programming, end-user application programming, information processing, data visualization, instruction, documentation, and maintenance all in one software environment.

Within the context of scientific database management systems, there are numerous examples of visual interfaces for geographical information systems or global change research [5, 7]. These systems improve user interfaces by allowing direct-manipulation of spatially related information. The programming languages used are still textual. In [3, 14], some database issues related to geographic information systems are addressed and visual techniques are used for query specification.

The focus of all the above systems are restricted to only parts of a complete scientific data analysis system. To clearly present the deficiencies of these systems, the conceptual model of scientific data analysis and its supporting requirements have to be investigated.

2 Scientific Data Analysis and Supporting Requirements

The process of scientific data analysis has been widely studied [16, 11, 21]. One generalized model of data analysis, defined by Hick [11], involves four phases: collect and process data, set up a computational model, execute the computational model, and interpret the results. The procedural description of the model is shown in Figure 1. The real research process may repeat these phases many times before a satisfactory result or conclusion is obtained.

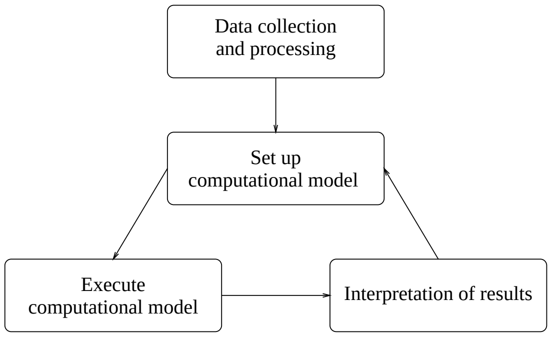

Figure 1: Life-Cycle of Data Analysis [11].

Further analysis of this model has led us to the development of an enhanced conceptual framework for data analysis. We refer to a single iteration of the data analysis task in Figure 1 as an *experiment*. The purpose of an experiment is to generate new knowledge about a discipline from the data and knowledge that are available. We call the computational model in the second phase of Figure 1 a *process*. A process can be described by a flow diagram as shown in Figure 2. In a process flow diagram, two kinds of nodes are placed alternately. Operator nodes represent the analysis or processing operations applied to data. Data nodes represent information used or generated by operators. A more detailed description, based on a Petri net model of data, can be found in [10].

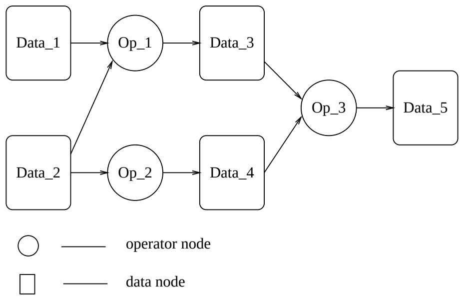

Figure 2: Diagram of an Analysis Process.

A process is actually just a template; the data types, operators, and flow of execution are specified, but not the physical data sets to be used. The combination of a process with a specific set of data defines an experiment. It is clear that a relationship exists between experiments which share the same processes but differ in the data sets used. Similarly, different processes can generate data which are related at some level of abstraction. We refer to this relationship as a *concept*, and each data set in our system is associated with a particular concept.

Considering the elationships between these components of a data analysis experiment, the data analysis model is modified as shown in Figure 3. A data model consisting of concepts and data objects is generated and modified in a data analysis experiment. The interpretation of results is based on this model and controls the iteration of the the experiment.

Based on our study of the data analysis model, several observations may be made:

- In an experiment, the role of processes is as important as the role of data. In a process, the role

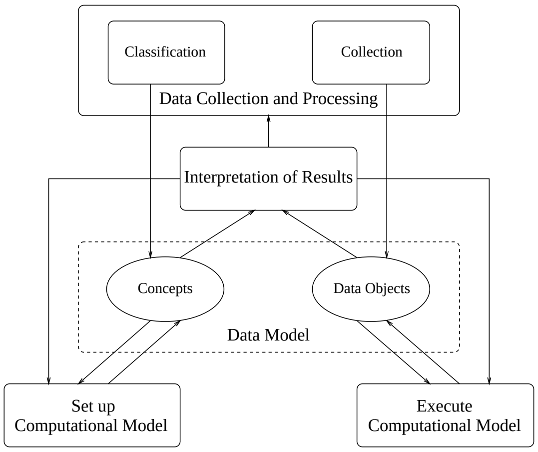

Figure 3: A Modified Data Analysis Conceptual Model.

of operators is as important as the role of data.

- There are always concepts behind data, but without interpretation, concepts behind data cannot be seen. The process applied to original data defines the semantics (concepts) of the final output data of that sequence.

- Experimentation is an iterative process. Data generated from an experiment can be used to generate new concepts and enhance old concepts.

From the above observations, we can develop specific requirements for supporting data analysis through an easy-to-use user interface:

- The interface should clearly represent concepts, data, processes, and experiments.

- The interface should provide users intuitive ways of manipulating information, including the management of data, processes , and experiments.

- Generating and executing an experiment should be easy, so iteration and repetition will be facilitated.

- Information regarding experiments, processes, concepts, operators, and data should be available at any stage of an experiment.

- Information should be able to be visualized for the purpose of interpretation.

Now let's look at the deficiencies with the visualization systems discussed in first section. First, information about data, processes and experiments are assumed to be understood by users. For example, selecting a process from the process library requires a fair level of sophistication from users, as they have to know what the processes do and what they are named. In some cases, when the experiment is small and the data used in the experiment is simple, this mode of operation is acceptable. But in a more general case, information may be shared among scientists, so more sophisticated methods of managing data, processes and experiments have to be provided. The second problem is that in most current systems, only processes are represented explicitly in the user interface. Data are specified simply as parameters to processes, and experiments and concepts are not supported at all. These deficiencies are being addressed in the design of the GaeaVE.

We take research in geographical information systems and global change as an example of scientific research. Our analysis of properties involved in this research supports our conclusions above. There are three additional significant properties of data in global change research [9]:

- Spatial Data

  Spatial data are one or more dimensional structured data. Spatial data types are not as simple as those found in conventional database systems. In [8], spatial data are defined as n-dimensional data with explicit knowledge about objects, their extent and position in space. Such an object can be a city that is made up of streets and buildings. Data representing a city have a molecular structure. Spatial data introduces a diverse set of data types, such as points, line segments, and regions.

  Since spatial data have a more complex structure than conventional data, the properties of spatial data and their relationships are more complex. The operators and predicates which characterize those properties and relationships are, of course, richer. The operations on conventional data consist of numerical computation or alphanumeric comparison, and the only relationships supported are equality and ordering. Spatial data relationships and operations include containment, overlay, neighborhood, and distance.

- Temporal Data

  Data with temporal attributes is the basis of global change research. The understanding of time differs among scientists [18]. It is hard to give a definition of temporal data without defining a particular point of view. Temporal data structures and operators can only be discussed after a temporal data model is created. One possibility is to treat temporal data in the same manner as one-dimensional spatial data. In this model, time is defined as a

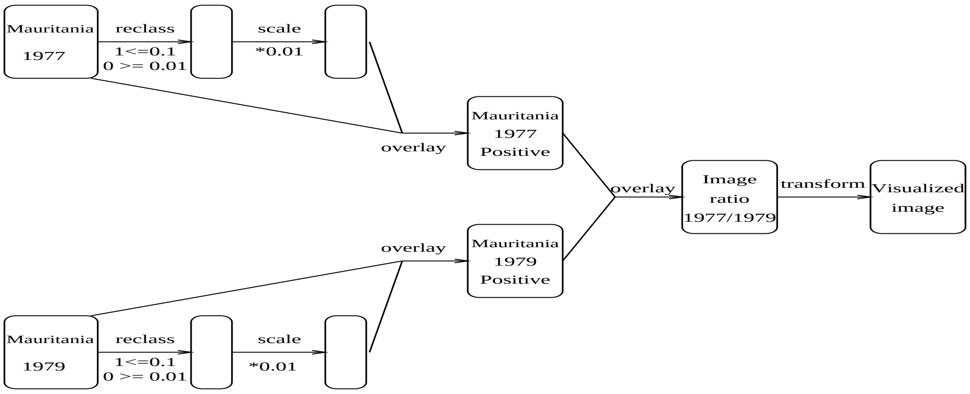

Figure 4: An Example of an Experiment: Image Ratioing.

set of points or intervals, and operators analogous to those found in spatial data may be used.

- Diverse Data

  Another property of data in GIS and global change research is that spatial data, non-spatial data, temporal data, and non-temporal data are all involved [6]. This encourages us to consider spatial data, temporal data, and attribute data in a consistent and unified fashion rather than independently.

As an example of a scientific experiment in global change research, Figure 4 shows the process of image ratioing [4]. We are only interested in the process that a data analysis follows. The details of this process applied to global change research are presented in [4]. In this work, NDVI (Normalized Digital Vegetation Index) data is the original concept. The images from 1977 and 1979 for an area of Mauritania along the Senegal River are instances of this concept, and ratioing is the process. The objective is to illustrate a new concept (NDVI ratio) and an instance of this concept (for Mauritania in 1977/79) after applying a process of some pairwise comparison techniques using data on a large scale. The analysis process is as follows.

1. Process NDVI data of both 1977 and 1979 to consist of only positive values. The operators used include *reclassify, scale*, and *overlay.*

2. Apply functions on data generated in Step 1 to do ratioing. The operators used are *overlay, image ratio*, and *transform.*

3. Apply more functions on data generate in Step 2 to examine data. The operators used are *examine, reclassify*, and *visualize.*

4. Describe the results or conclusions based on the examination in Step 3.

3 The GaeaVE System

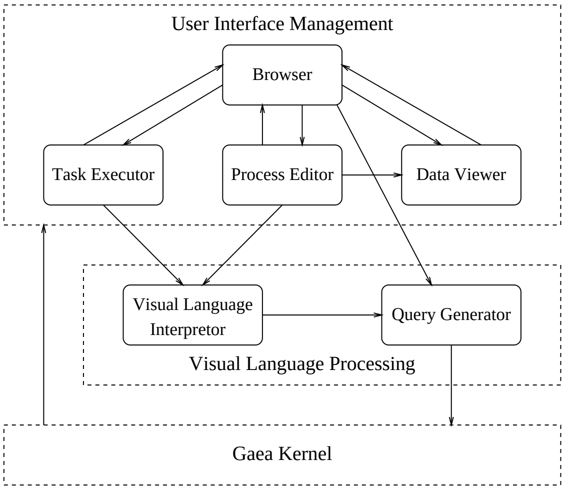

Figure 5: The GaeaVE System Organization.

The GaeaVE system organization is shown in Figure 5. The current version of GaeaVE can be summarized as follows:

- Use different visual variables (eg. color, icons and windows) to differentiate between different components of scientific data analysis (i.e. concepts, data, processes, experiments, etc.) and represent the relationships between those components.

- Provide a programming environment that interprets scientists' data analysis activity intuitively and handle both data and process management requests.

- Provide a browser that can be driven according to users' conceptual models to retrieve information stored in the database.

- Provide special mechanisms for the specification of spatial and temporal objects and relations.
- Provide a consistent view of all components across different levels of abstraction and phases of operation.

3.1 Visual Browsing

The browsing system supports retrieval of all information stored in the application database system. GaeaVE supports multiple ways of information browsing.

- Browsing data and process semantics

  The derivation semantics of data is defined by the process that is used to generate that data. On the other hand, the semantics of a process are defined by the input data and output concepts for that process and the operators it employs. From this point of view, browsing methods include identifying data available to be used by a process and identifying processes available to work on a piece of data. An example is shown in Figure 6.

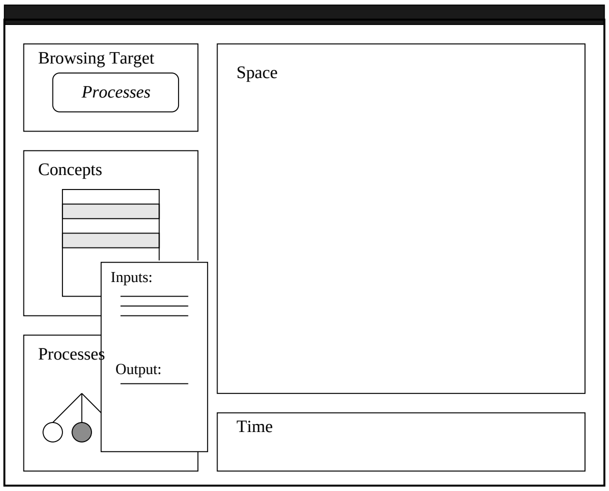

Figure 6: Semantic Browsing: The input and output information is in a popup window and the data available is highlighted in the concept window.

- Browsing data or concepts in time and space

  Data has multiple aspects: spatial, temporal and attribute. For viewing these aspects, browsing is accomplished by specifying one or two of the three aspects and displaying what, if any, data are available meeting the specified constraints. Thus the user can determine, for example, what types (attributes) of data are available for a given region and period of time by identifying the spatial and temporal extents. An example is shown in Figure 7.

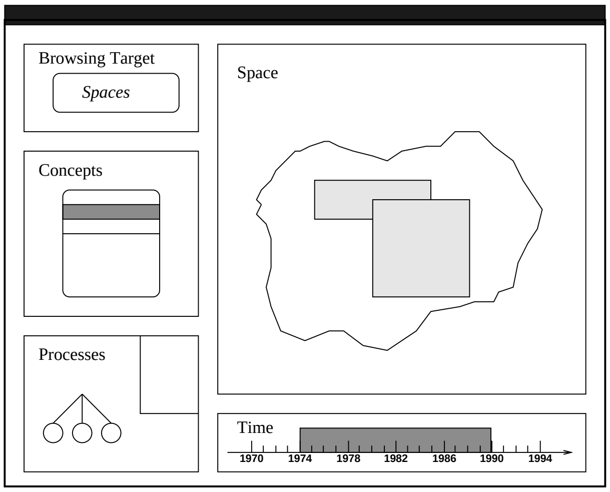

Figure 7: Spatial Browsing: The result of a browsing request is highlighted in the space window, the conditions are specified in other windows using a different color.

- Browsing data and processes in the conceptual model

  Besides the derivation semantics, concepts in the application conceptual model can drive users' browsing activities. Typical questions to ask in the browser include how a concept is defined, and what data and process are associated with a certain concept.

3.2 Visual Analysis Programming

The GaeaVE's programming environment has two major components: a process editor and a task (experiment) executor. In the process editor, a process can be defined or modified, while in the task executor, an experiment can be set up and executed.

Both processes and experiments are specified using data flow diagrams as the programming language. A data flow programming language is more suitable than other programming language styles, since data flow diagrams on paper are frequently used by scientists before any data analysis via the computer [2, 12, 15, 17]. One major difference between the GaeaVE data flow language and most other data flow languages is that data as well as processes are represented as nodes. Different node representations are used for data nodes and process nodes. This way, data and processes take equivalent positions in the appearance of an experiment. Users can switch their points of interest between data and processes easily. Figure 8 shows the visual program for the global change research experiment example shown earlier.

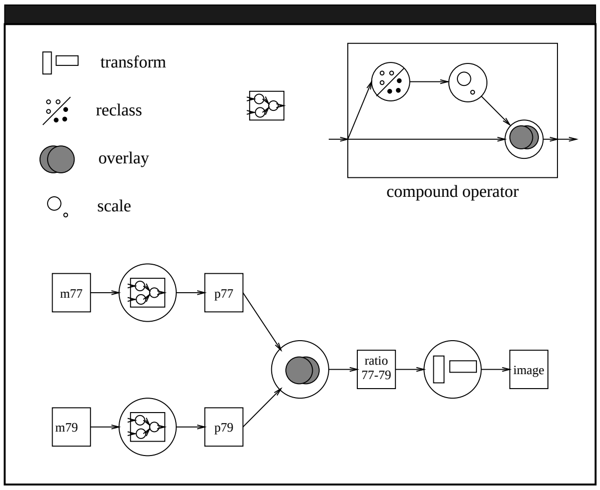

Figure 8: A Program Written in the Gaea Visual Language.

In a data-flow diagram, whenever all the input data for a process has arrived or an input data is updated, the process will be executed and will generate output to all the nodes to which it is connected. In GaeaVE, this can be done both automatically (as in AVS) or manually as users desire.

In the GaeaVE programming environment, separating the process editor and the task executor is to distinguish between processes and experiments. In both the process editor and the task executor, common components, such as a particular data object or attribute type, are represented using the same icon.

In the GaeaVE browser, different window areas and icons are used for data, processes, space, time, and concepts. The general use of visual variables is summarized below:

- Icons or shapes are used to differentiate between distinct components. For example, in the task executor, a rectangular icon represents a data class and a circular icon represents a process. Icon bitmaps are designed to convey object semantics, and a key to interpretation is available on demand.

- Color is used to indicate the difference between instances of a concept. For example, in one concept, different data instances are colored differently. Color is also used in the process editor to associate attributes of a concept. Attributes of the same color are properties of the concept sharing that color.

- Shading is used to represent the existence of a data object. A shaded icon represents an existing data object. A nonshaded icon means that either the data object does not exist or the data object has not been sufficiently specified to refer to a unique data object. Shading is also used for processes to indicate if a process is ready to execute. A shaded process has all input data to execute, An unshaded process is still expecting more data.

- Flashing animation is used to show the execution of a process or experiment.

A common problem of visual languages is that the screen space is so limited that not all information can be shown. Generally, in scientific research, the average size of the data flow diagram will fit in a reasonably sized computer screen. Even in large experiments, scientific research activity usually focuses on one part of a detailed process rather than on the whole experiment. GaeaVE uses strategies such as sliding windows and iconification to increase the utilization of the screen.

3.3 Data Visualization

For data visualization, the GaeaVE provides a data viewer. Data with various types, such as tabular data, image data, and geometry data, can be viewed using a set of visualization techniques. Users can interactively specify the parameters for visualization operators. The results of visualization can be used for specifying spatial extents or location, data types, or temporal intervals or points for use by the Browser.

Since much of the data that Gaea deals with have a spatial property, the GaeaVE provides a general support for representing spatial data. Maps are the traditional tools used by global change researchers to represent spatial data. In recognition of this, spatial specification in GaeaVE is accomplished via map representation and manipulation. However, simply generating a map representation from an image or a vector form is insufficient to support scientists' research activities. More sophisticated map manipulation tools are supported. For each map display, the set of manipulation functions includes:

- Selecting – Getting a point or region in image or world coordinates.

- Rescaling – Selecting display at different scales.

- Repositioning window – Selecting display for locations of interest.

- Changing Attributes – Changing the display appearance by specifying display variables, such as color, symbol icons, etc.

- Overlaying – Displaying two or more maps for the same area in a single display.

Figure 9 shows a map displaying political boundaries.

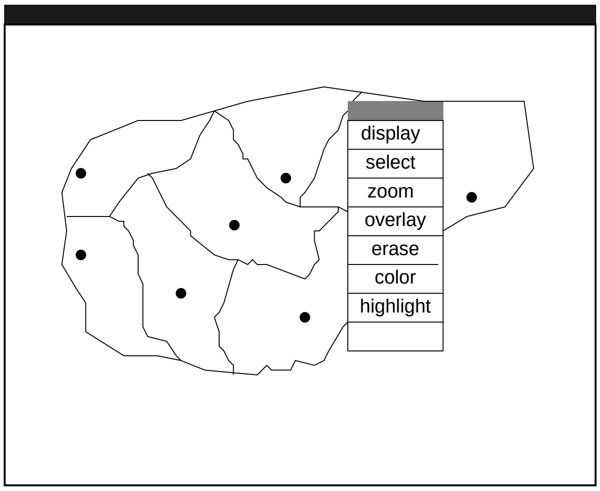

Figure 9: Spatial Data Representation.

The temporal aspect in data is also important in global change research and is one of the critical issues in the Gaea project. It is the responsibility of the GaeaVE to support visualization of change and other time-related properties. The Gaea system treats time analogously to one-dimensional spatial data. According to this abstraction, time is represented as a one-dimensional axis in the GaeaVE. functions include:

- Selecting – Selecting time point (a point on display) or time period (a segment on display).
- Rescaling – Changing granularity of time.
- Repositioning – Changing the time period being displayed.

4 Research Status and Conclusion

At present, each of the components of the GaeaVE has been prototyped and independently tested. Current efforts are focussed on establishing communication links between the independent components so that, for example, the browser may pass information regarding data or operators to the process editor and attributes from a data visualization may be used by the browser to specify new queries to the database. Work is also underway at developing methods for capturing, managing, and displaying semantics (concepts) from experiments.

In this paper, we have outlined our view of the data analysis and management needs for scientific data analysis in the global change research field and our progress towards addressing these needs. The Gaea system, currently under development, integrates the management of experiments, processes, data, and operators in a single environment. We believe the Gaea Visual Environment contributes significantly towards the requirements of scientific data analysis by supporting visual programming at both the process level and experiment level, presenting clear visual representations of data, processes, and concepts, and providing a powerful database browser.

References

[1] Advanced Visual System Inc., 300 Fifth Avenue, Waltham, MA 02154. *AVS Technical Overview*, October 1992.

[2] J. J. Birchman and S. L. Tanimoto. An implementation of the viva visual language on the next computer. In *1992 IEEE Workshop on Visual Languages*, pages 177–183, Seattle, Washington, September 1992.

[3] E. Clementini, A. D'Atri, and P. Di Felice. Browsing in geographic databases: an object-oriented approach. In *1990 IEEE Workshop on Visual Languages*, pages 125–131, Skokie, Illinois, October 1990.

[4] J. R. Eastman and J. E. McKendry. *Explorations in Geographic Information Systems Technology.* UNITAR European Office, Switzerland, 1991.

[5] M. J. Egenhofer. Manipulating the graphical representation of query results in geographic information systems. In *1990 IEEE Workshop on Visual Languages*, pages 119–124, Skokie, Illinois, October 1990.

[6] A. U. Frank. Properties of geometric data: Requirements for spatial methods. In *Advances in Spatial Databases, 2nd symposium, SSDSD'91*, Zurich, Switzerland, August 1991. Spring-Verlag.

[7] P.-C. Goh. A graphical query language for cartographic and land information systems. *SIGMOD RECORD*, 19(4):61–68, December 1990.

[8] O. Guenther and A. Buchmann. Research issues in spatial database. *SIGMOD RECORD*, 19(4):61–68, December 1990.

[9] N. I. Hachem, M. A. Gennert, and M. O. Ward. A DBMS architecture for global change research. In *Proceedings of ISY Conference on Earth and Space Science*, Pasadena, CA, 1992.

[10] N. I. Hachem, K. Qiu, M. A. Gennert, and M. O. Ward. Managing derived data in the Gaea Scientific DBMS. Worcester Polytechnic Institute Computer Science Department Technical Report Number WPI-CS-TR-08, 1992.

[11] C. R. Hicks. *Fundamental Concepts in the Design of Experiments.* Holt, Rinehart and Winston, New York, NY, 1982.

[12] Daniel D. Hils. Visual languages and computing survey: Data flow visual programming languages. *Journal of Visual Languages and Computing*, 3(1):69–101, March 1992.

[13] Y. E. Ioannidis, M. Livny, and E. M. Haber. Graphical user interfaces for the management of scientific experiments and data. *SIGMOD Record*, 21(1):47–53, March 1992.

[14] E. Jungert. Towards a visual query language for an object-oriented geographical information system. In *1990 IEEE Workshop on Visual Languages*, pages 132–137, Skokie, Illinois, October 1990.

[15] J. Kodosky, J. MacCrisken, and G. Rymar. Visual programming using structured data flow. In *1991 IEEE Workshop on Visual Languages*, pages 34–39, Kobe, Japan, October 1991.

[16] P. D. Leedy. *Practical Research: Planning and Research.* Macmillan Publishing Co. Inc, New York, NY, 1980.

[17] Thomoas J. Olson, Nicholas G. Klop, Mark R. Hyett, and Shawn M. Carnell. Mavis: A visual environment for active computer vision. In *1992 IEEE Workshop on Visual Languages*, pages 170–176, Seattle, Washington, September 1992.

[18] K. Qiu, N. I. Hachem, M. O. Ward, and M. A. Gennert. Providing temporal support in data base management systems for global change research. In *Proceedings of SSDM 92*, Switzerland, 1992.

[19] N. C. Shu. *Visual Programming.* Van Nostrand Reinhold Company, New York, NY, 1988.

[20] Silicon Graphics Computer Systems, Mountain view, CA. *IRIS Explorer*, 1991.

[21] Rebecca R. Springmeyer, Meera M. Blattner, and Nelson L. Max. A characterization of the scientific data analysis process. In *1992 IEEE Workshop on Visual Languages*, Seattle, Washington, September 1992.

[22] Craig Upson, Thomas Faulhaber, David Kamins, David Laidlaw, David Schlegel, Jeffrey Vroom, Robert Gurwitz, and Andries van Dam. The application visualization system: A computational environment for scientific visualization. *IEEE Computer Graphics and Applications*, 9(4):30–42, July 1989.

[23] Carla. S. Williams and John. R. Rasure. A visual language for image processing. In *1990 IEEE Workshop on Visual Languages*, pages 86–91, Skokie, Illinois, 1992.