AD-A256 862

USACERL Technical Report EC-92/01
August 1992
Fundamental Database Processes

US Army Corps
of Engineers
Construction Engineering
Research Laboratory

Predicting Database Requirements for Geographic Information Systems in the Year 2000: Long-Term Design Issues for GRASS

by
Kurt A. Buehler
Michael Shapiro
James D. Westervelt

The Environmental Compliance Modeling and Simulation Division of the U.S. Army Construction Engineering Research Laboratories is responsible for continuing development of the geographic information system (GIS) known as GRASS—the Geographic Resources Analysis Support System. As the use of GRASS has increased, several significant limitations in the system have been identified. At the same time, advances in hardware and software technology have occurred that might help to address some of these limitations. This report presents the results of a preliminary investigation into the software capabilities and implementation strategies involved in revising the GRASS GIS. The research showed that one promising way to enhance GRASS would be to use an extended relational database management system or an object-oriented database management system as the core of the new GIS.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1992	Final

4. TITLE AND SUBTITLE

Predicting Database Requirements for Geographic Information Systems in the Year 2000: Long-Term Design Issues for GRASS

5. FUNDING NUMBERS

PE 4A161102
PR AT23
WU SA-X01

6. AUTHOR(S)

Kurt A. Buehler, Michael Shapiro, and James D. Westervelt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Construction Engineering Research Laboratories (USACERL)
PO Box 9005
Champaign, IL 61826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER

EC-92/01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Construction Engineering Research Laboratories (USACERL)
PO Box 9005
Champaign, IL 61826-9005

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Environmental Compliance Modeling and Simulation Division of the U.S. Army Construction Engineering Research Laboratories is responsible for continuing development of the geographic information system (GIS) known as GRASS—the Geographic Resources Analysis Support System. As the use of GRASS has increased, several significant limitations in the system have been identified. At the same time, advances in hardware and software technology have occurred that might help to address some of these limitations. This report presents the results of a preliminary investigation into the software capabilities and implementation strategies involved in revising the GRASS GIS. The research showed that one promising way to enhance GRASS would be to use an extended relational database management system or an object-oriented database management system as the core of the new GIS.

14. SUBJECT TERMS

Geographic Resources Analysis Support System
GRASS
Geographic Information Systems

15. NUMBER OF PAGES

44

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

FOREWORD

This study was conducted by the U.S. Army Construction Engineering Research Laboratories (USACERL) under Project 4A161102AT23, "Basic Research in Military Construction"; Work Unit SA-X01, "Fundamental Database Processes."

This research was performed by the Environmental Compliance Modeling and Simulation Division (EC) of the Environmental Sustainment Laboratory (EL), USACERL. William D. Goran is Chief, CECER-EC. Dr. Edward W. Novak is Acting Chief, CECER-EL. The USACERL technical editor was Gordon L. Cohen, Information Management Office.

COL Daniel Waldo, Jr., is Commander and Director of USACERL, and Dr. L.R. Shaffer is Technical Director.

CONTENTS

PREDICTING DATABASE REQUIREMENTS FOR GEOGRAPHIC INFORMATION SYSTEMS IN THE YEAR 2000: LONG-TERM DESIGN ISSUES FOR GRASS

1 INTRODUCTION

Background

In 1983 the U.S. Army Construction Engineering Research Laboratory (USACERL) Environmental Division (EN) created the Geographic Resources Analysis Support System (GRASS), a computer-based geographic information system (GIS). Since that time GRASS has been under continuous development to address the Army's increasingly complex land analysis needs.

Over the years, the use of GRASS has increased in the public and private sectors to assess environmental impacts, evaluate site suitability, detect change over time, manage cultural, historical, and natural resources, and model the effects of environmental phenomena. These increased demands on GRASS have resulted in the recognition of several limitations in the system as currently implemented. Limitations in the software include the lack of support for aspatial data associated with spatial entities, the lack of a user-friendly graphical interface, the lack of vector analysis capability, and the inability to store anything other than integer data. Some of these limitations could be addressed by upgrading the current GRASS system or by integrating other software in a loose manner. A better approach might be to redesign and develop a new generation of GRASS that addresses these limitations while preserving the successes of the current systems.

The initial design of GRASS in the early 1980s targeted hardware that would be available in the near-term future—5 to 10 years—and such hardware did become popular in the workplace. Machines now on the drawing board will bring more power and new capabilities at a lower cost to the workstation community to which GRASS will remain targeted. These hardware environments will open wide the doors to greater software potentials. As software has long been limited by hardware capability, these potentials are also defined and limited by underlying data structures and data storage techniques. It is important, therefore, to define such structures and techniques in a way that allows for the most effective exploitation of the hardware environment.

Objective

The objective of this study was to develop recommendations for addressing some of the weaknesses in current GISs—specifically in GRASS—including discussion of how the recommendations might affect software capabilities and possible implementation strategies for achieving those capabilities.

Approach

The authors reviewed the recent literature in the area of next-generation database management systems (including object-oriented database management systems [OODBMS] and third-generation or extended relational database management systems [RDBMS]). In addition, literature in four areas of software capability were examined, especially in reference to GISs:

- Image processing systems
- Spatial modeling and analysis
- Map data input, output, and editing
- User interfaces.

Literature detailing the application of advanced hardware to these areas of capability was examined. Limitations and problems that would most likely present themselves when implementing these new capabilities were identified.

The research is presented in two parts: (1) software capabilities and requirements, including a description of one possible future GIS environment for reference, and (2) implementation strategies. Several areas of necessary research were identified.

Scope

This report presents results of a preliminary investigation into the problems discussed in the Background section. It is not intended to answer every question related to correcting the limitations of GRASS, but instead presents one possible solution, implementation strategies, and recommended directions for further research.

Mode of Technology Transfer

The information presented in this report is being evaluated for incorporation into future versions of GRASS. Resulting technologies will be implemented in the software targeted at the general GRASS user community. Technology transfer will also be supported through the updating of user documentation and services offered through the GRASS Information Center at USACERL.

2 GIS SOFTWARE CAPABILITIES AND REQUIREMENTS

The Need for Innovation

GISs require the storage of large amounts of data which must be provided to multiple users on demand. GIS users come from varying backgrounds and, thus, have differing data manipulation needs. Cartographers, for example, might want to combine information from several sources to construct maps that convey a desired message. Land managers may use the data to develop large tracts of land for management plans based on the characteristics of the land and the activities being planned. Environmental scientists may model physical phenomena using GIS data as inputs, and display functionality to visualize results. Engineers may want to integrate the models developed by scientists into simulations both of interactions between physical processes and effects of these interactions on their designs. Individuals in many fields may wish to investigate previously undiscovered relationships between different spatially referenced data. From these few examples it is obvious that a GIS must provide a wide range of functionality. To effectively apply GISs to their problems, users need new, innovative systems that will provide them with the tools for constructing, navigating, and processing geographic databases.[1]

Additionally, aspatial data is commonly used in conjunction with spatial data. An environment that supports all of the activities cited in the previous paragraph must also handle all types of data and the relationships among those data. Next-generation applications (e.g., computer-aided design and manufacturing systems, multimedia and hypermedia information systems, expert systems) require databases that can support the requirements of these applications.[2] GISs fall into this category, and it is obvious that future GIS environments must be based on object-oriented database management systems.

Perhaps the most interesting plight is that of the engineer trying to model interaction between physical systems using a GIS. Since this is among the most demanding kind of work engineers currently do on GISs, the current research has focused on ways to improve GISs to facilitate such complex modeling tasks. In the following pages, the authors outline an idealized system that might address the specific weaknesses of current GISs. This idealized system is offered as a frame of reference for subsequent discussions throughout this report.

One Possible Future GIS Organization

A GIS must consist of a toolbox that includes functional modules[3] that operate on specific data types (or "objects"). This requirement stems from the fact that engineering tasks frequently can be boiled down to operations on complex physical objects that can be expressed in terms of their state variables, parameters, and operations on these variables and parameters. Examples of such a modular concept might include:

- A module for performing algebra on a set of raster images
- A module for performing a shortest path analysis on a network of arcs and nodes
- A module for combining raster data, with vector linework and textual annotations for the construction of cartographically correct maps

---

[1] Dutton, G., "Improving Spatial Analysis in GIS Environments," *Auto-Carto 10* (March 1991), pp 168-185.

[2] Joseph, J.V., et al., "Object Oriented Databases: Design and Implementation," *Proceedings of the IEEE*, 79(1) (Institute of Electrical and Electronics Engineers [IEEE], January 1991), pp 42-64.

[3] Westervelt, J., "The Two Classes of GIS Users (Or How to Make Software Salad)," *GIS World*, 3(5) (October 1990), pp 111-112.

- Modules for extending the set of data types to include problem-specific and complex data objects. (An example here would be a watershed object made of several component objects such as reservoir objects, stream-reach objects, diversion objects, waste treatment plant objects, and so on. Multiple instances of these objects would commonly be connected to form a specific watershed being modeled.)

To implement this concept the GIS of the future should be constructed on top of an extended RDBMS or an OODBMS that can handle many types of data, both spatial and aspatial.[4] Figure 1 depicts one possible organization of such a GIS. This organization is layered, with each layer building on the one below and providing the functionality needed to build the next layer. The underlying layer of DBMS software is accessible via an application programming interface (API) and/or a query interface (QI). If users want to browse the database before or during application execution, the query interface provides this function. Functional modules are then constructed using the API to store and retrieve data. These functional modules are used in applications (APP) along with the API to form the next level of access. Finally, the graphical user interface (GUI) level is on top. Note that the user interface level has access to applications, functional modules, and the QI. The following paragraphs describe the four levels of use according to the type of person accessing that level.

The End User Level. From the standpoint of the end user, this GIS would allow both direct queries of the geographic database and the ability to graphically invoke application programs. This level of user might see a palette of icons (an "icon manager") representing applications and functional program modules used to operate on data. Each application or functional module would have its own interface for gathering information about input "sources," output "sinks," and application parameters.

The Application Programmer Level. The applicable programmer would use the GIS at the end user level to create and register data flow applications for the end users who use macros. The macros would be created using a regular text-based editor or a visual programming environment (VPE). The VPE would allow application programmers to assemble functional modules in a data flow arrangement, and register the completed "pipeline" (application) with the icon manager. An example of a VPE is depicted in Figure 2. Assume an application is desired that consists of applying functions X, Y, and Z consecutively to an input map and then displaying the output map. The application programmer need only drag icons representing each of the five operations (input raster map, function X, function Y, function Z, and display raster map) from the icon palette into the visual programming area, connect them with rubber-banding lines that represent the flow of data between the modules, edit parameters associated with each module so the modules perform as expected (in an editor not shown), and press a start button in the control panel (not shown) to initiate the application for testing.

The Module Programmer Level. The module programmer would use an application programming interface to the database to create new functional modules using a high-level programming language, and may incorporate embedded query language code directly into application code as a cleaner interface to database functionality. This violates the level modularity presented here somewhat, but as long as new functionality added at level 4 is supported, then the movement of this functionality upward through the levels can be supported. Modules created at this level can then be integrated into the VPE for use by level 2 users.

The System Programmer Level. The system programmer would design new database types and create methods to operate on these types, thus extending the functionality of the database. This person would also be responsible for creating access to new functionality (methods) via the API and the embedded spatial query language interfaces for use by level 3 users.

The modules expected in such an environment should include the following categories of capability: (1) image processing, (2) modeling and analysis, and (3) data input, output, and editing. Each of these

---

[4] Frank, A.U., "Requirements for a Database Management System for a GIS," *Photogrammetric Engineering and Remote Sensing*, 54(11) (November 1988), pp 1557-1564.

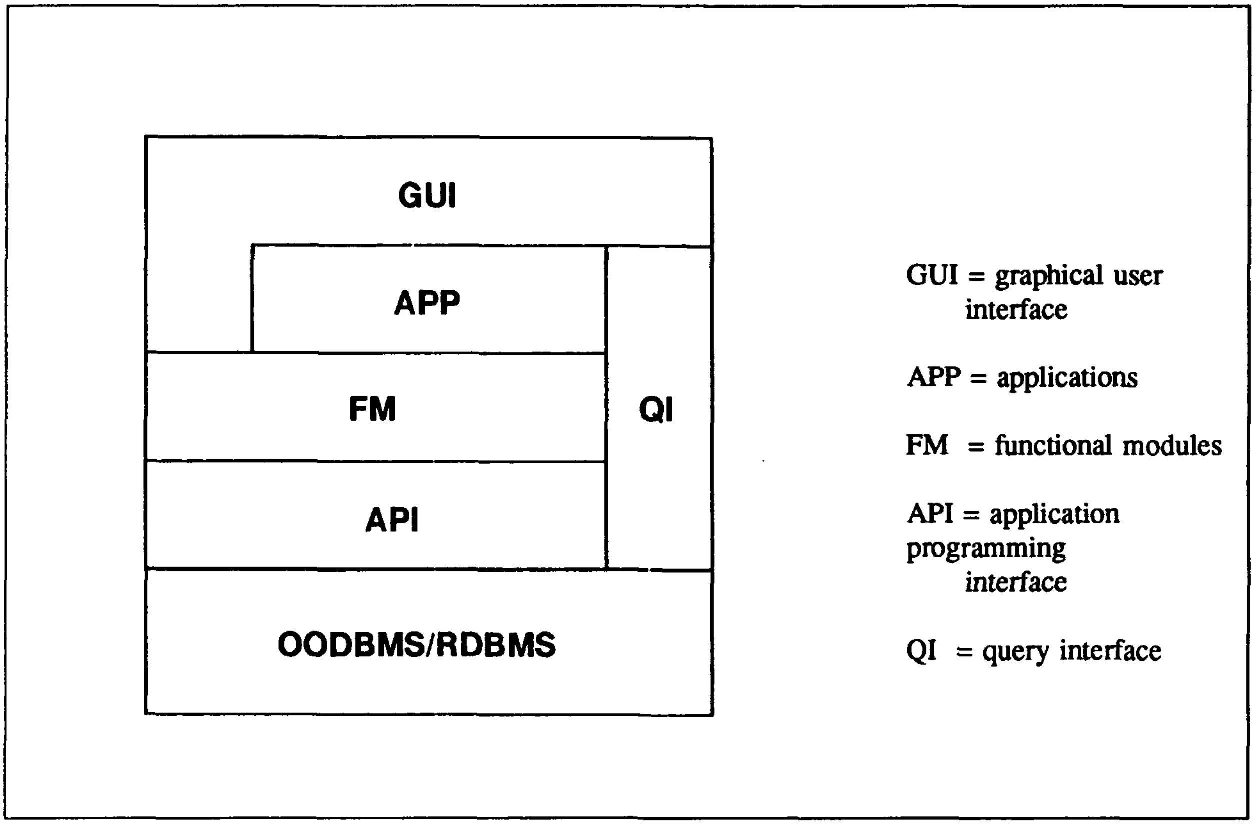

Figure 1. One Possible Organization of a Future GIS.

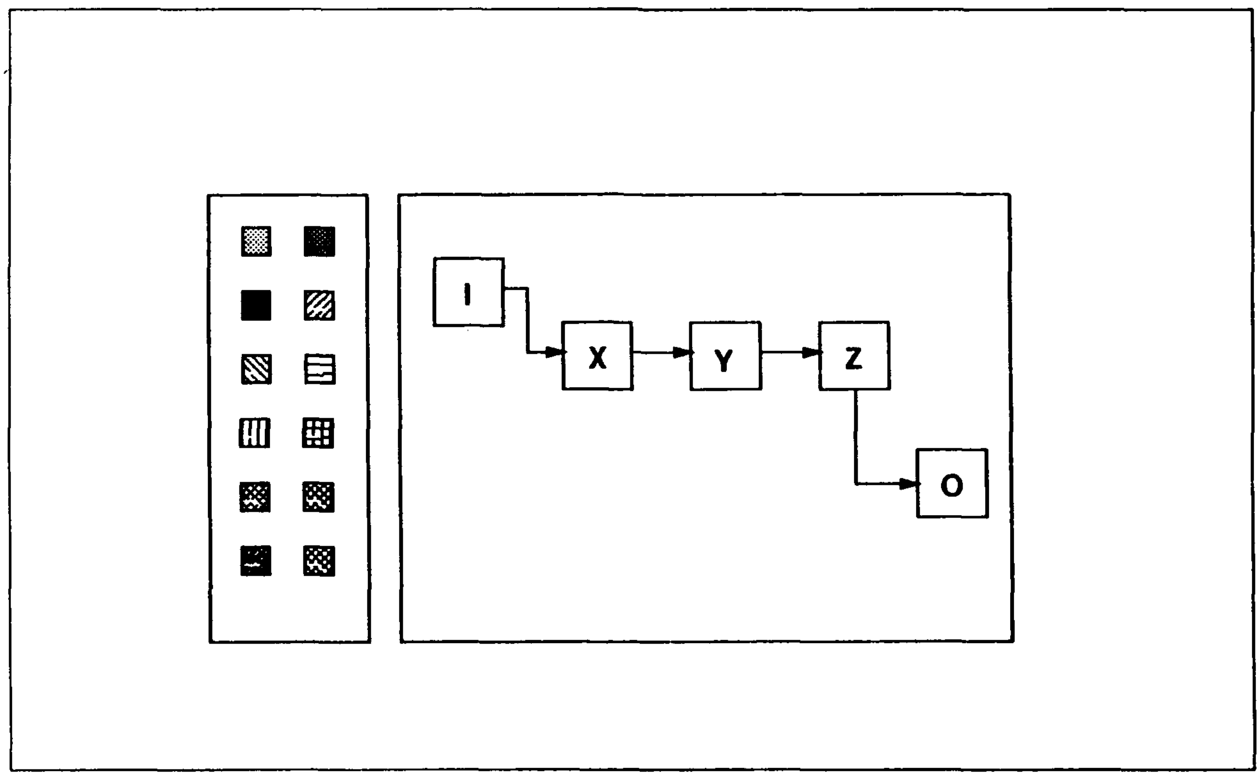

Figure 2. A Visual Programming Environment.

categories of functionality is discussed in the following sections. The next section details the software components described above.

Database Software

Clearly, one very important component of GIS software is that dealing with storage and retrieval of the data. Equally important are the data structures used to implement this software.[5] Many approaches could be taken in applying database software to a GIS. GIS functionality could be built around a custom DBMS, or the GIS database could be designed and written from scratch using specialized storage and retrieval techniques (GRASS is currently implemented the latter way). Both of these implementation strategies and their implications are discussed in Chapter 3. Several key areas identified merit further study and could help guide the development of future GIS database software. They are discussed below.

Object-Oriented and Extended Relational Database Systems

Currently, RDBMSs provide a wide variety of services, but in general they fail to fully support applications where the data are more complex, such as a GIS. DBMSs currently provide a fixed set of basic data types (character, integer, floating point, dollar amount, date, etc.) and a few functions to operate on them. Much research has been (and is) examining this difficulty in light of the increasing number of applications requiring a more capable DBMS.[6] Much research has concerned complex object (often referred to as non-first normal form, or NF^2) databases.[7] Complex object databases are either an extension of the relational database paradigm[8] or are created by building objects out of subobjects in an object-oriented fashion.[9] Using these models it is possible to model complex structured objects. These databases have mainly been targeted for application to computer-aided design (CAD) and computer-aided manufacturing (CAM) databases.[10]

Much research has also focused on the confluence of the object-oriented programming paradigm and the databases which has resulted in OODBMS.[11,12,13,14,15,16,17] Essentially, these systems merge the object-oriented language paradigm with those of database systems. OODBMSs do not follow the

---

[5] Cowen, D.J., "GIS versus CAD versus DBMS: What are the Differences?," *Photogrammetric Engineering and Remote Sensing*, 54(11) (November 1988), pp 1551-1555.

[6] Wilms, P.F., P.M. Schwarz, H.J. Schek, and L.M. Haas, "Incorporating Data Types in an Extensible Database Architecture," *Proc. 3rd Int. Conf. Data and Knowledge Bases: Improving Usability and Responsiveness* (June 1988), pp 180-192.

[7] Schek, H.J., et al., "The DASDBS Project: Objectives, Experiences, and Future Prospects," *IEEE Trans. on Knowledge and Data Engineering* (March 1990), pp 25-43.

[8] Dadam, P., et al., "A DBMS Prototype to Support Extended NF^2 Relations: An Integrated View on Flat Tables and Hierarchies," *Proc. 1986 ACM SIGMOD* (May 1986), pp 356-367.

[9] Harder, T., H. Schoning, and A. Sikeler, "Parallel Query Evaluation: A New Approach to Complex Object Processing," *IEEE Database Engineering, Vol 8* (1989), pp 21-29.

[10] Wilkes, W., P. Klahold, and G. Schlageter, "Complex and Composite Objects in CAD/CAM Databases," *Proc. 5th Int. Conf. Data Engineering* (February 1989), pp 443-450.

[11] Fishman, D.H., et al., "Overview of the IRIS DBMS" in W. Kim and F.H. Lochovsky, eds., *Object-Oriented Concepts, Databases, and Applications* (Addison Wesley, Reading, MA, 1987), pp 219-249.

[12] Wilkinson, K., P. Lyngbæk, and W. Hasan, "The IRIS Architecture and Implementation," *IEEE Trans. on Knowledge and Database Engineering*, 2(1) (March 1990), pp 63-75.

[13] Deux, O., et al., "The Story of O_2," *IEEE Trans. on Knowledge and Data Engineering*, 2(1) (March 1990), pp 91-108.

[14] Kim, W., et al., "Architecture of the ORION Next-Generation Database System," *IEEE Trans. on Knowledge and Data Engineering, 2(1)* (March 1990), pp 109-124.

[15] Agrawal, R., and N.H. Gehani, "ODE (Object Database and Environment): The Language and the Data Model," *ACM SIGMOD Record* (June 1989), pp 34-43.

[16] Boral, H., et al., "Prototyping Bubba, A Highly Parallel Database System," *IEEE Trans. on Knowledge and Data Engineering*, 2(1) (March 1990), pp 4-23.

[17] Haas, L.M., et al., "Starburst Mid-Flight: As the Dust Clears," *IEEE Trans. on Knowledge and Data Engineering*, 2(1) (March 1990), pp 143-159.

relational model and, therefore, do not typically include functionality commonly available in RDBMS (such as structured query language [SQL], although new query languages are being explored for these systems[18]). OODBMs are cited as having advantages for CAD/CAM databases, artificial intelligence applications, and office information systems.

Capabilities that have been explored include: rich data modeling constructs, direct database support for inference, novel data types (especially graphic images, voice, text, vectors, matrices), persistent data types (i.e., data that keeps its value across database sessions), long transactions (those lasting minutes to days), and multiple versions of data. The OODBMS has been shown to be superior to the RDBMS in that the schema is superior: it is capable of providing convenient access to information, and it is easier to extend and maintain.[19]

Some very interesting research has been done by Stonebraker and Rowe[20] with the design of POSTGRES, the successor to the INGRES relational database system.[21] According to Stonebraker and Rowe, the main design goals of POSTGRES are to:

1. Provide better support for complex objects

2. Provide user extensibility for data types, operators, and access methods

3. Provide facilities for active databases and inferencing, including forward- and backward-chaining

4. Simplify the DBMS code for crash recovery

5. Produce a design that can take advantage of optical disks, workstations composed of multiple tightly coupled processors, and custom designed VLSI* chips

6. Make as few changes as possible—preferably none—to the relational model.

The benefit of the POSTGRES model over the others is that all of the useful characteristics of the relational model are retained while exploiting the benefits of the object-oriented model. Clearly, the first two goals listed above satisfy most of the modeling needs of engineers, but the remaining goals *satisfy the need to take into account future hardware technologies*. Furthermore, because POSTGRES source code is in the public domain, its usefulness for specific research into geographic application is enhanced. Recently, a large project called Sequoia 2000 was initiated at several campuses of The University of California; one of this project's goals is to integrate GRASS with POSTGRES.[22] Although it is not the ultimate goal of Sequoia 2000, this integration should provide important insights into the problems with building a GIS with a RDBMS core.

---

[18] Blakeley, J.A., C.W. Thompson, and A.M. Alashqur, "Strawman Reference Model for Object Query Languages," *X3/SPARC/DBSSG/OODBTG Task Group Workshop* (October 1990).

[19] Ketabchi, M.A., S. Mathur, T. Risch, and J. Chen, "Comparative Analysis of RDBMS and OODBMS: A Case Study," *Proc. IEEE COMPCON 90* (February 1990), pp 528-535.

[20] Stonebraker, M., and L. Rowe, "The Design of POSTGRES," *IEEE Database Engineering*, Vol 6 (1987).

[21] Stonebraker, M., et al., "The Design and Implementation of INGRES," *ACM Trans. on Database Systems*, 1(3) (September 1976), pp 189-222.

* VLSI: very-large-scale integration.

[22] Stonebraker, M., and J. Dozier, *Sequoia 2000: Large Capacity Object Servers to Support Global Change Research*, Technical Proposal (University of California, May 1991).

While it appears that the work of the POSTGRES team may have its advantages, it is not clear that POSTGRES will be able to process the large amount of data required in a raster GIS with the level of performance desired by users. Also, the advantages offered by the object-oriented paradigm are not explicitly available in the POSTGRES system as of version 3.0. The reader is referred to several papers of interest and insight by the POSTGRES development team.[23,24,25,26,27,28,29]

Distributed Database Technology

Distributed database technology is important because users of computer systems in general are becoming decentralized. In other words, they are relying more than ever on sharing data over networks. As long as there exists a network of homogeneous databases that somehow can find each other, users can have truly distributed databases. This is a very powerful concept because it allows local sites to have data that are used routinely in a local database while allowing less-used databases to remain at the sites where they are most needed.[30] Data distribution will prevent the need for costly replication of geographic databases.

As an aside, it should be mentioned that the "open systems" concept—very loosely defined as the provision of system architecture specifics needed to integrate or move software across platforms—is going forward and the possibility of interoperability among heterogeneous systems is imminent. The International Standards Organization's Open Systems Interconnection (ISO-OSI)[31] and Open Software Foundation's (OSF) Application Environment Specification (AES),[32] to name only two, are indications that there is significant interest in standardizing interconnection and user environments. This interest indicates that open systems will become a reality in the future. What this may mean from a database standpoint is that database software will be able to provide direct, read-only access to Oracle®, for example, or IBM Database 2 (DB2) databases, whether residing locally or on remote systems. The major concerns here are probably how to deal with concurrent access, network permissions, security, and how to deal with long transactions that might be interrupted, but these problems are currently being studied. In addition, and most importantly, there are institutional constraints that would effectively prevent the sharing of data in the first place.

Spatial Query Languages

The implications of allowing a user to directly query the database, as opposed to controlled queries via interfacing programs, should be studied in detail. Having said this, the user may find it helpful to be able to make a custom query directly on the database instead of having a programmer somehow develop an interface to the database software. A user might wish to be able to directly query the database for

---

23 Stonebraker, M., and J. Dozier, May 1991.
24 Stonebraker, M., "Inclusion of New Types in Relational Database Systems," *Proc. 2nd Int. IEEE Conf. Data Engineering* (February 1986).
25 Stonebraker, M., "The POSTGRES Storage System," *Proc. 12th Very Large Database Conference* (IEEE, 1987).
26 Stonebraker, M., et al., "Extensibility in POSTGRES," *IEEE Database Engineering*, Vol 6 (IEEE, 1987).
27 Rowe, L., and M. Stonebraker, "The POSTGRES Model," *Proc. 12th Very Large Database Conference* (IEEE, 1987), pp 83-96.
28 Greene, D., "An Implementation and Performance Analysis of Spatial Data Access Methods," *Proc. 5th Int. IEEE Conf. Data Engineering* (February 1989), pp 606-615.
29 Stonebraker, M., L. Rowe, and M. Hirohama, "The Implementation of POSTGRES," *IEEE Trans. on Knowledge and Data Engineering*, 2(1) (March 1990).
30 Kameny, I., "Global Information System Issues," *Proc. 5th Int. IEEE Conf. Data Engineering* (February 1987), pp 672-673.
31 Zimmerman, H., "OSI Reference Model—The ISO Model of Architecture for Open Systems Interconnection," *IEEE Trans. on Communications*, Com-28(4) (April 1980), pp 425-432.
32 Open Software Foundation, *Application Environment Specification* (Prentice-Hall, 1990).

normal analysis purposes. For example, a user should be able to give a generic command like "create new raster = good_sites where soils = [1,3,7,9-12] and slope < 5 and land_use != [2,3,4,7-9] and distance from roads < 200 meters."

The usefulness of such a query mechanism seems obvious. Indeed, extensive work has been done concerning spatial queries, and the reader is referred to Greene[33] or, for a more extensive view, to Samet.[34] As for the usefulness of SQL as a basis for a spatial query language, this matter is probably best viewed by observing that systems using SQL have become successful in the past. This is most likely because SQL and languages like it are perhaps intuitive, because they are English-like, and their meaning can be interpreted in most cases by reading the query. Clearly, an SQL-like language would require special extensions that would make sense only in the context of a GIS. For an excellent discussion of the pros and cons of SQL, see articles by Stonebraker[35] and Frank.[36]

Raster-Vector Integration

Well researched and proven data structures (or classes of data structures) that are better suited to both basic types of geographic data already exist. To try to generalize and adapt one generic data structure to these data types would preclude the storage and manipulation efficiencies provided by the separate data types. By retaining the optimal data structure for each data type, the benefits of these structures can be exploited. The problem with this approach is that specific techniques are required to access, manipulate, and display each data type. The real trouble comes when the data type to be stored is complex or interrelated with other data (although the POSTGRES system mentioned earlier provides some of this capability). Issues and suggestions relating to data structures are discussed further in Chapter 3.

Image Processing

Image processing has received much attention in the literature lately, especially in the application of parallel processing hardware. This was no doubt driven for the most part by the field of computer vision because of the need to process images in real time.[37,38] It has been shown many times that images can be processed much faster using parallel hardware and even using parallel software techniques on sequential processors.[39,40,41]

Another technology of interest for application to the image processing tasks—especially pattern recognition tasks—is the neural network. Limited research has been done on integrating an image processing system with neural networks.[42]

---

[33] Greene, D., pp 606-615.

[34] Samet, H., *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS* (Addison-Wesley, Reading, MA, 1990).

[35] Stonebraker, M., "Future Trends in Data Base Systems," *IEEE Trans. on Knowledge and Data Engineering*, 1(1) (March 1989), pp 33-44.

[36] Frank, A.U.

[37] Rice, T.A., and L.H. Jamieson, "Parallel Processing for Computer Vision" in S. Levialdi, ed., *Integrated Technology for Parallel Image Processing* (Academic Press, 1985).

[38] Dew, P.M., R.A. Earnshaw, and T.R. Heywood, eds., *Parallel Processing for Computer Vision and Display* (Addison-Wesley, NY, 1989).

[39] Dew, P.M., et al.

[40] Duller, A.W.G., R.H. Storer, A.R. Thomson, and E.L. Dagless, "An Associative Processor Array for Image Processing," *Image and Vision Computing*, 7(2) (May 1989), pp 151-158.

[41] Page, I., ed., *Parallel Architectures and Computer Vision* (Clarendon Press, 1988).

[42] Wu, X., and J.D. Westervelt, *Engineering Applications of Neural Computing: A State of the Art Survey*, Special Report N-91/22/ADA237628 (U.S. Army Construction Engineering Research Laboratory [USACERL], May 1991).

Apart from the fact that image processing will take advantage of parallel architectures is the question of the software that would drive the application. Integrated image processing software packages abound, both in the public domain and especially in the private sector as vendors scramble to meet the needs of industry and government.[43,44,45] Possible new image processing software for GIS application might include modules for automatic image classification, automatic feature extraction (for automatically extracting certain data and possibly for removal of cartographic annotations in scanning paper maps), change detection and analyses, etc. Also possible are fast implementations of the standard set of image processing algorithms for geometric correction, image enhancement and filtering, and classification.

Image processing for GIS application should be able to directly use advances in image processing for other applications. In fact, the authors believe most image processing software could be applied directly because there is no need for the software to recognize a geographic reference in order to perform its tasks. The user need only supply the data as a matrix and let the database software handle the locational issues.

Modeling and Analysis

This area of software is key to the successful application of GIS technology by engineers modeling physical systems. If the appropriate tools are available to the engineer, a successful application of GIS technology can be realized. However, some areas are lacking in current GISs, and these must be addressed.

As stated previously, an engineer can model a system using objects. The modeler can tailor an object to have all of the characteristics and behavior needed to model the system in question. The database software can help in this by facilitating the creation of objects and corresponding methods that act on the object and communication between objects. Once the modeler has implemented the objects needed to model a system, there must be a mechanism for simulating the behavior of the system over time. The database software can again come to the aid of the modeler by supplying rules (or "triggers") and timers that can be associated with the data objects. Triggers monitor the database for some conditions and they are associated with objects. Triggers make it possible to perform actions within the database without actively using the database. Once a trigger condition is met, the trigger action–usually some form of database transaction—is scheduled for completion. In this way, the database becomes active and triggers can start a flurry of activity without the presence of an operator. Timers can also be used in conjunction with rules or triggers to provide a simulation mode. Timers can be tied to real wall-clock time or they can be used to generate virtual time increments. Timers can vary in "granularity," meaning that a timer click can be viewed as anything from a microsecond to a millennium.

Software for performing many tasks in modeling and analysis of engineering data is available currently. Software developed in this category will be driven in large part by the applications that will be funded for development. Hydrologic modeling applications will probably appear first, as there seems to be a great interest in solving these problems within a GIS framework.[46]

---

[43] Brown, V., and A.M. Gemazian, "The March of GIS," *Workstation News*, 2(2) (February 1991), p 24.

[44] Harmon, S., "The Image Makers," *Workstation News*, 2(2) (February 1991), p 22-23.

[45] Warnick, L., and D. Blaylock "Theory of Revolution," *Workstation News*, 2(2) (February 1991), pp 26-27.

[46] Johnson, L.E., et al., "Geographic Information Systems for Hydrologic Modeling," *Proc. 3d ASCE Water Resources Operations Management Workshop: Computerized Decision Support Systems for Water Managers* (June 1988), pp 736-749.

Map Data Input, Output, and Editing

Input of geographic data using digital tape, optical disks, scanners, tablet digitizers, and even video should be available, and input by data conversion from other formats should also be available. Sophisticated output of the data should also be facilitated by supporting plotters, printers, multiple concurrent displays, and video.

The GIS should also be able to convert data into other formats to allow transfer of data between systems. The user should be able to create new data or modify existing data through some form of editing (on-screen digitizing) capability. This editing facility might even include a CAD-like capability with the ability to edit all geographic data types. This would be a useful capability, considering that much digital cartography is simply electronic drafting.[47] Extension of this editing capability to GIS would present problems as well as solve them. This capability would be facilitated by database software that could handle complex data objects.[48] Import and export facilities that could interface with popular CAD formats would therefore be desirable.

User Interface

It is likely that all of the software predicted in this report will employ sophisticated GUIs. The proliferation of user interface development products and research indicates that the industry expects GUIs to be a very important component of future software systems. In fact, GISs have been criticized heavily because they are traditionally difficult to learn, use, and customize. Furthermore, user proficiency gained on one product is not easily transferred to another.[49] The user interface is the appropriate place to "glue" all of a system's disparate modules together into a cohesive whole. There is little doubt that GUIs are the "way of the future," if one can judge by the number of articles on the topic in recent trade journals.

The X Window System™ is a network-based graphics system developed by MIT and adopted as an industry standard. It is expected that most GUIs developed for Unix-based GISs in the future will use the X Window System, probably with a user interface such as OSF/Motif.

Many GISs available today are difficult to learn, use, and customize. Furthermore, experience gained from using one GIS is not readily transferrable to the use of another. These problems have been addressed by UGIX,[50] an interface environment that is independent of the underlying GIS. UGIX could be used as the basis for a more standardized user interface if widely accepted by the GIS user community.

GISs tend to be large collections of program modules. It is often difficult to combine these modules into a coherent new application. Visual programming environments such as those found in the XVision image processing system[51] and the animation production Environment (apE) visualization software[52] are becoming very popular because they allow users to create applications by graphically arranging program modules on the screen. VPEs use icons (or "glyphs") to represent independent program modules. These icons are then connected graphically in terms of data flow to build specific applications.

---

47 Cowen, D.J.

48 Wilkes, W., et al.

49 Raper, J.F., and M.S. Bundock, "UGIX: A GIS Independent User Interface Environment," *Auto-Carto 10* (March 1991), pp 275-295.

50 Raper, J.F., and M.S. Burdock.

51 Rasure, J., S. Hallett, and R. Jordan, "A Comprehensive Software System for Image Processing and Programming," *SPIE Proceedings Vol. 1075, Digital Image Processing Applications* (January 1989), pp 37-45.

52 Anderson, H.S., et al., *The Animation Production Environment: A Basis for Visualization and Animation of Scientific Data*, Ohio Supercomputer Graphics Project Technical Report TR-04 (Ohio State University, March 1989).

It is likely that the GIS of the future will incorporate at least the following features within its GUI:

- An easy-to-use API
- A set of command-line programs built with the API
- A VPE for building applications based on the command-line programs
- A GUI including an overall interface to all GIS functions and a tool for performing direct database queries (as described previously).

The GUI described in the last point above should integrate functionality provided by the API and the functionality contained in the command line (and as extended by the VPE).

The API will be built directly on the DBS, allowing low-level access to data and database functionality. The ability to include in application programs embedded queries, similar to the query language, would also be a positive feature.

3 IMPLEMENTATION ISSUES AND STRATEGIES

As stated earlier, the GIS functionality described in this report could be built into a custom DBMS, or the GIS database could be designed and written from scratch using specialized storage and retrieval techniques designed specifically for geographic data types. Before beginning a discussion of the pros and cons of each strategy, the authors will present some background that should be kept in mind during that discussion. Because any next-generation GIS must address anticipated advances in hardware technology, it is important to mention some interesting technological issues, and their relationship to GIS.

Hardware and Software Issues

This section describes several important issues concerning the implementation of advanced GIS software on advanced hardware.

The Need for High Performance

The application of GIS technology to engineering modeling tasks has been very limited. One may wonder why, considering that the GIS model fits well with many engineering modeling tasks. The answer may simply be that the hardware platforms upon which GISs are implemented do not supply the "horsepower" needed for many engineering tasks, but actually it is more likely to result from a combination of factors. Hardware speed is certainly one of these factors. Another is that GISs, as currently implemented, simply do not provide the functionality required for many engineering models. One key function missing from GISs is the ability to effectively add a temporal dimension to a model. This shortcoming could be addressed by implementing rules and timers in the GIS database (as discussed in the previous chapter under "Database Software" and "Modeling and Analysis").

The answer to this question also depends on the applications faced by the GIS user. For real-time or pseudo real-time applications, or for computation-intensive procedures, high performance hardware platforms are a must. However, if rapid gains in sequential machine performance continue at the present pace, it may be unnecessary to parallelize all but the most compute-intensive algorithms.

Parallel Architectures

In an attempt to offer some insight into the application of parallel processing technology to GISs, three basic categories of parallel computer architecture are described below: SIMD (single instruction multiple data), MIMD (multiple instruction multiple data), and hybrid SIMD/MIMD architectures.

SIMD machines represent a form of synchronous, highly parallel processing.[53] Systems with as many as 216 processors are currently available commercially. An SIMD machine consists of a control unit, a set of processing elements (PEs), each with its own memory, and an interconnection network. The control unit broadcasts instructions to all PEs, and each active PE executes the instruction on the data in its own memory. The interconnection network allows data to be transferred among the PEs. This arrangement is well suited for exploiting the parallelism inherent in certain tasks performed on vectors and arrays. As SIMD architectures mature, the speed of nearest-neighbor and inter-PE communication will reach very acceptable levels. In fact, configurations exist where intercommunication rates have already exceeded 1 gigabyte per second. One problem with the SIMD configuration is the transfer of data from

[53] Duncan, R., "A Survey of Parallel Computer Architectures," *IEEE Computer*, 23(2) (February 1990), pp 5-16.

disk to processors and vice versa (input/output subsystem). Input/output (I/O) rates have reached 1 gigabyte per second in systems with specially designed hardware, however.

MIMD machines represent asynchronous parallel processing.[54] MIMD systems with many thousands of processors are currently available, although the number of processors is usually under a 100. A MIMD machine usually consists of P processors and M memories, M greater than or equal to P, where each processor can follow an independent instruction stream. Two major categories of MIMD architecture are shared memory and local memory.

In *shared-memory* MIMD architectures the PEs share a common global memory, or have access to both global and local memory. In *local-memory* (or "distributed memory") MIMD architectures, each PE has its own local memory, and data are transferred via an interconnection network much like an SIMD architecture. MIMD architectures are useful for general parallel processing tasks. In terms of GIS implementation, the main disadvantage of MIMD architectures is that computation is asynchronous. Many GIS computations must be synchronous, however, because they must share data and, thus, rely on the completion of certain computations by other processors. This synchronization problem leads to programming difficulties and does not take full advantage of the capability of the architecture.

Most commercially available parallel machines are actually a hybrid of SIMD and MIMD architectures. At the extreme end of the hybrid spectrum are truly configurable SIMD/MIMD machines like the PASM[55] (partitionable SIMD/MIMD) system. The model assumed here combines SIMD and MIMD attributes. Each PE contains the same code but executes the code on different data. However, within each PE, the code can run in MIMD mode. This modification of the basic model allows faster execution on some code than in the pure SIMD model without the problems inherent to the full flexibility of an MIMD machine. This architecture has been proven successful in practice for computer vision, image processing, and pattern recognition tasks.

There is no way to know with certainty which, if any, of the architectures outlined above will actually be available in a low-cost workstation by the year 2000, but one can make some reasonable projections based on industry trends. SIMD architectures are currently available for under $200,000 (with up to 1024 processors), but it seems unlikely that this architecture will be common in workstations because it is well suited for only a relatively small class of applications. However, it will be necessary to monitor progress in the area of SIMD hardware. Relatively inexpensive add-on hardware may possibly become available. It is more likely that MIMD architecture will end up in the mainstream. In fact, several vendors currently offer machines with eight or more processors for under $150,000, and it seems likely that the more general-purpose MIMD architecture will be more readily available. In fact, now available are add-on hardware devices called transputers, which each contain one or more PEs. When connected to a workstation, transputers can make a MIMD machine out of an existing workstation at a reasonable cost. In addition, software that is now available allows the user to configure a network of workstations to act as a single MIMD machine.[56] The SIMD/MIMD hybrids, although well suited to the GIS framework, are primarily (and will probably remain) research tools, and will probably not be available on a production basis for many years.

---

[54] Duncan, R.

[55] Siegel, H.J. et al., "PASM: A Partitionable SIMD/MIMD System for Image Processing and Pattern Recognition," *IEEE Trans. on Computers*, C-30(12) (December 1981), pp 934-947.

[56] Duncan, R.

Technological Bottlenecks

Several bottlenecks that currently inhibit performance are now being overcome, which will allow the successful application of parallel processing technology to GISs.[57] Processor speeds are improving at a dramatic rate. However, there is more to a computer system than the processor. Gains in memory and I/O bandwidth must match the gains due to higher processor speeds and use of more processors in order to have a system in which the performance potential of the processors is realized. Although significant problems remain, memory and I/O rates are reported to be reaching very acceptable limits.[58]

If a set of processors needs data to perform some operation, it must usually access some kind of secondary memory (usually magnetic disk) to bring the desired data into PE memory. One possible solution to this problem is to provide some type of data or file cache, where frequently used data can be stored off-PE until needed. This could work because memory access speeds are much higher than disk transfer speeds. Data could be cached in memory, reducing the need for disk access. Toward this end, dynamic random access memory (DRAM) technology is advancing. This technology is currently expensive, but the cost is constantly decreasing. This DRAM storage of data (sometimes referred to as solid state disk, or SSD) will become more and more feasible as the cost per megabyte of storage continues to decrease.[59]

Meanwhile, other technologies that address the I/O bottleneck are being explored. Read/write optical disk technology is maturing rapidly, and data transfer rates now exceed 1 megabyte per second. As the cost for this technology decreases, it may prove useful when combined with disk arrays or log-structured file systems.

Given the similar performance and cost per megabyte of large and small disks, one way to improve performance is to replace a single large drive by an array of smaller drives.[60,61] Such an array provides higher performance because many small requests can be serviced independently and large requests can be spread over several disks to transfer in parallel. This technique, known as data striping, is especially useful in systems where large data transfers are a frequent requirement, such as a GIS. Log-structured file systems append the most recent additions to the end of a log. This can reduce disk seek times.[62]

Some set of I/O bottleneck solutions in tandem with MIMD architecture, is the most likely approach to be employed in the low-cost, highly parallel workstation of the year 2000. It also seems very likely that GISs will be implemented on such a platform to take advantage of the increased performance.

---

[57] Gaudiot, J.L., "Parallel Computing: One Opportunity, Four Challenges," *Proc. 5th Int. IEEE Conf. Data Engineering* (February 1989), pp 482-484.

[58] Frieder, O., "Communications Issues in Data Engineering: Have Bandwidth—Will Move Data," *Proc. 5th Int. IEEE Conf. Data Engineering* (February 1989), p 674.

[59] Bate, G., "Alternative Storage Technologies," *Proc. IEEE COMPCON 89* (February 1989), pp 151-157.

[60] Meador, W.E., "Disk Array Systems," *Proc. IEEE COMPCON 89* (February 1989), pp 143-146.

[61] Patterson, D.A., et al., "Introduction to Redundant Arrays of Inexpensive Disks (RAID)," *Proc. IEEE COMPCON 89* (February 1989), pp 112-117.

[62] Douglis, F., and J. Ousterhout, "Log-structured File Systems," *Proc. IEEE COMPCON 89* (February 1989), pp 124-129.

Notwithstanding the fact that the architectures themselves are "moving targets," many researchers (and programmers) have pointed out the difficulties associated with software development on parallel architectures in general.[63] Potential solutions to the problem of programming complexity include:

- Development of architecturally independent high-level languages (e.g., Linda,[64] Seymour[65])
- Extensions of existing languages (e.g., Fortran 8x, C*, *Lisp)
- Vectorizing compilers
- Developing automated tools for the detection of parallelism in existing codes.

The authors believe that, as research into these areas continues, the programmability problem will be solved at least enough to make the task of developing software manageable. It is also predicted that the solution will probably employ architecturally independent high-level languages.

Two Implementation Strategies

The following sections discuss two implementation strategies that could be taken to realize the GIS and software described in Chapter 2. Discussion will also address how GRASS fits into these scenarios.

Development of a Specialized Storage and Retrieval System

The methodology in this strategy would be to develop a new storage and retrieval system based solely on the geographic data types deemed necessary to implement and the specialized data structures found to be best for this task. The data types of interest would be those usually associated with GISs: raster and vector (the latter including areal, linear, and pointal features). Many of the limitations discussed in Chapter 2 under "Raster-Vector Integration" would still need to be addressed, as well as a few more discussed below.

Systems employing this strategy are now starting to appear. The Environmental Systems Research Institute (ESRI) has just released the latest version of its GIS. In addition to its traditional vector-based functionality, this GIS now also includes raster-based functionality and methods for transferring data between the types. Earth Resources Data Analysis System (ERDAS) has recently released ERDAS-ARC/INFO Live Link. Both of these systems, as well as others not mentioned here, are available now. Both seem to be quality systems and, through the INFO portion of ARC/INFO, have the ability to handle aspatial data. The major drawbacks to these systems are (1) cost to the user, (2) the fact that they are hard to learn and use, and (3) inflexibility due to a lack of customization and access to internal data structures. These systems are currently limited to visual integration; that is, they are really not yet integrated at the data structure level.

Indeed the great advantage of GRASS, besides its superiority for raster analysis, has been its relatively small cost and the direct access it offers to internal data structures for customization and special purpose applications. With these points in mind, the development needs of GRASS may be explored to discuss how the system's weaknesses might be remedied by a specialized storage and retrieval system. GRASS's principal development needs are the following:

- The ability to handle floating point data
- An improved and more complete vector capability

---

[63] Lewis, T.G., "Issues in Parallel Programming: Why Aren't We Having Fun Yet?" *Supercomputing Review* (July 1990).

[64] Carriero, N., and D. Gelernter, "Linda: Some Current Work," *Proc. IEEE COMPCON 89* (February 1989), pp 98-101.

[65] Miller, R., and Q.E. Stout, "An Introduction to the Portable Parallel Programming Language: Seymour," *Proc. IEEE COMPSAC 89* (September 1989), pp 94-101.

discuss how the system's weaknesses might be remedied by a specialized storage and retrieval system. GRASS's principal development needs are the following:

- The ability to handle floating point data
- An improved and more complete vector capability
- An improved handling of aspatial data.

To address the first point above, the handling of floating point data could be accomplished by modifying the current raster data structure to handle both integer and floating point data. The specific methodology involved in making this extension would probably be to place a flag in the raster header file designating the raster map as type integer or type floating point, and to include the design of a new storage and retrieval function for floating point maps. There still exists the problem of how to actually store floating point numbers in a machine-dependent way. This problem could be solved by selecting the most common floating point format and providing special routines for converting between other formats.

To address the second point, vector capability could be improved to include the kind of analysis capabilities found in systems such as ARC/INFO.

The third point, handling aspatial data, is a twofold problem. First, it requires the improvement of vector capability such that it will be more straightforward to handle attributes that are spatially related to a map's features. Secondly, it requires some mechanism for storing and relating aspatial data. This second task has traditionally (and correctly) been approached by integrating commercial DBMS software such as Oracle or INGRESS. It might be advantageous to explore integration with OODBMS or extended RDBMS as discussed in Chapter 2.

This implementation strategy still has several drawbacks. Existing raster-based GRASS applications would have to be rewritten to accommodate the floating point data type. This strategy would also require modification of all existing GRASS vector software and creation of completely new vector functionality. In addition to these drawbacks, there is the psychological barrier of going through the trouble to implement something that is basically already available—or available with much less effort—in existing database management systems.

Also, this approach would leave the GIS developers at "Square 1" in making engineering modeling tasks easier to implement. Perhaps the most bothersome aspect of this strategy is that it gives no attention to the projected advances in hardware technology. To accommodate these advances, completely new software would have to be developed.

Using an Extended RDBMS or OODBMS Core

This methodology would be to use an extensible DBMS, such as POSTGRES or a public domain OODBMS, as the core of a GIS as described in Chapter 2. Using a DBMS as the basis for the GIS would offer all of the advantages inherent in these systems:

- *Better support for complex objects.* This would enhance the ability to model physical systems, as discussed in Chapter 2 under "Modeling and Analysis."

- *The provision of extensible data types (including multiple inheritance).* This would include the ability to define operators and access methods for these new data types. It would provide the mechanism for implementing raster and vector data types by simply extending existing types. It would also provide the ability to create specialized data types based on the basic geographic types. The ability to define spatial operators would make data manipulation easier to deal with and also could provide for the inclusion of high level topological relationships such as "within", "beside", "above", or "near". The ability to define access methods would allow for optimization by making it possible to define various access methods that might prove to make certain tasks simpler to implement.

- *The ability to have active databases and inferencing.* Both of these features would add many new potential capabilities to applications that would be developed using such a system. These include a database history through multiple versions of data, triggers, and timers.

- *The ability to take advantage of advances in hardware.* If the GIS could exploit technologies such as optical disks, multiple tightly-coupled processors, and custom chips, less effort in this direction would be necessary when developing new applications for this system. A major criterion in selecting EDMBS or OODMBS would be its inherent support for advanced hardware technologies.

- *Strong support for aspatial data.* This is inherent in the relational database model.

The tasks left to accomplish would be to define the new spatial types, operators, and access methods, and to implement existing tools using the new database. The authors do not intend to minimize the difficulty in completing these tasks, but the points outlined above represent some major advantages to using a core DBMS in the next generation of GRASS.

Research Issues for a DBMS-Based GIS

The use of a DBMS as the basis of the GIS seems to be the best way to provide for many of the shortcomings of today's systems, while reducing the effort required to overcome system weaknesses and take advantage of hardware that is on its way from the laboratory to the market.

The observation and judgments presented here are based on research into the current state of GIS technology and projected developments in low-cost workstation hardware. The next logical step would be to test the hypotheses developed here by implementing a prototype system. In this connection there is another important advantage to using a DBMS as the core of the GIS: prototyping is facilitated by the DBMS itself. The cost of aborting the idea due to unforseen circumstances is much reduced. The authors believe the most promising approach would be to implement a prototype using an object-oriented programming language and an OODBMS in order to take advantage of the efficiencies associated with object-oriented programming. Ultimately the research must determine whether an OODBMS would or would not be more effective as the core of the GIS.

Specific research questions that must be answered early in the research and development process include the following:

- Does an OODBMS provide adequate support for the large amounts of data that must be stored in a GIS?

- Does an extended RDBMS provide adequate support for large amounts of GIS data?

- What is the best way to deal with raster (or image) data within an object-oriented framework?

- Assuming that both an extended RDBMS and an OODBMS can support the data requirements for a GIS, what are the advantages and disadvantages of each with regard to spatial analysis and modeling tasks?

- How do OODBMS and extended RDBMS support (or fail to support) projected advances in hardware technology, and how will they promote or hinder the use of these technologies?

- How do extended RDBMS and OODBMS differ in their ability or inability to support raster databases?

- How can OODBMS facilitate the representation of high level topological relationships?

These questions can most effectively be answered through development of a DBMS-based prototype for USACERL's next-generation GIS.

4 SUMMARY

When USACERL created GRASS in the early 1980s, the GIS was designed for implementation on hardware that would be available up to 10 years into the future. Subsequently, the system was widely accepted among workstation users both in the public and private sectors. As the number of GRASS users (and uses) grew, new demands were put on the software, some of which highlighted limitations in the current system. These include:

- Lack of support for aspatial data associated with spatial entities
- Lack of a user-friendly GUI
- Lack of vector analysis capability
- The inability to store anything other than integer data.

Also, hardware development has proceeded at a rapid rate since GRASS was first introduced. GRASS is not able to take advantage of extra processing power available in new hardware now coming to market.

While some of these challenges could be met by upgrading the current system, a more effective approach would be to design a new generation of GRASS that addresses these issues while preserving the successful features of the current system.

The degree to which these issues may be addressed in the next generation of GRASS will be defined and limited by underlying data structures and data storage techniques. One possible design for the next generation of GRASS would consist of a "toolbox" of functional modules that can operate on specific data types. The system would be organized into layers of functionality corresponding to the domains of system programmer, the module programmer, the applications programmer, and the GRASS end user. Two possible approaches to implementing this new design were discussed:

- Development "from scratch" of a specialized storage and retrieval system exclusively for geographic data types

- Building the required GIS functionalities into an advanced, customized DBMS.

While the first strategy has been employed with some success in upgrades of other GISs, its chief drawback may be that it does not specifically give attention to projected advances in hardware technology.

The second strategy offers inherent benefits that meet important GRASS development needs (e.g., strong support for aspatial data, ability to exploit new hardware technology). The authors consider the potential problems with this strategy to be surmountable through research and development.

REFERENCES

Cited

Agrawal, R., and N.H. Gehani, "ODE (Object Database and Environment): The Language and the Data Model," *ACM SIGMOD Record* (June 1989), pp 34-43.

Anderson, H.S., et al., *The Animation Production Environment: A Basis for Visualization and Animation of Scientific Data*, Ohio Supercomputer Graphics Project Technical Report (March 1989).

Bate, G., "Alternative Storage Technologies," *Proc. IEEE COMPCON 89* (February 1989), pp 151-157.

Blakeley, J.A., C.W. Thompson, and A.M. Alashqur, "Strawman Reference Model for Object Query Languages," *X3/SPARC/DBSSG/OODBTG Task Group Workshop* (October 1990).

Boral, H., et al., "Prototyping Bubba, A Highly Parallel Database System," *IEEE Trans. on Knowledge and Data Engineering*, 2(1) (March 1990), pp 4-230.

Brown, V., and A.M. Gernazian, "The March of GIS," *Workstation News*, 2(2) (February 1991), p 24.

Carriero, N., and D. Gelernter, "Linda: Some Current Work," *Proc. IEEE COMPCON 89* (February 1989), pp 98-101.

Cowen, D.J., "GIS versus CAD versus DBMS: What are the Differences?," *Photogrammetric Engineering and Remote Sensing*, 54(11) (November 1988), pp 1551-1555.

Dadam, P., et al., "A DBMS Prototype to Support Extended NF^2 Relations: An Integrated View on Flat Tables and Hierarchies," *Proc. 1986 ACM SIGMOD* (May 1986), pp 356-367.

Deux, O., et al., "The Story of O_2," *IEEE Trans. on Knowledge and Data Engineering*, 2(1) (March 1990), pp 91-108.

Dew, P.M., R.A. Earnshaw, and T.R. Heywood, eds., *Parallel Processing for Computer Vision and Display* (Addison-Wesley, NY, 1989).

Douglis, F., and J. Ousterhout, "Log-structured File Systems," *Proc. IEEE COMPCON 89* (February 1989), pp 124-129.

Duller, A.W.G., R.H. Storer, A.R. Thomson, and E.L. Dagless, "An Associative Processor Array for Image Processing," *Image and Vision Computing*, 7(2) (May 1989), pp 151-158.

Duncan, R., "A Survey of Parallel Computer Architectures," *IEEE Computer*, 23(2) (February 1990), pp 5-16.

Dutton, G., "Improving Spatial Analysis in GIS Environments," *Auto-Carto 10* (March 1991), pp 168-185.

Fishman, D.H., et al., "Overview of the IRIS DBMS" in W. Kim and F.H. Lochovsky, eds., *Object-Oriented Concepts, Databases, and Applications* (Addison Wesley, Reading, MA, 1987), pp 219-249.

Frank, A.U., "Requirements for a Database Management System for a GIS," *Photogrammetric Engineering and Remote Sensing*, 54(11) (November 1988), pp 1557-1564.

Frieder, O., "Communications Issues in Data Engineering: Have Bandwidth—Will Move Data," *Proc. 5th Int. IEEE Conf. Data Engineering* (February 1989), p 674.

Gaudiot, J.L., "Parallel Computing: One Opportunity, Four Challenges," *Proc.* 5^{th} *Int. IEEE Conf. Data Engineering* (February 1989), pp 482-484.

Greene, D., "An Implementation and Performance Analysis of Spatial Data Access Methods," *Proc.* 5^{th} *Int. IEEE Conf. Data Engineering* (February 1989), pp 606-615.

Haas, L.M., et al., "Starburst Mid-Flight: As the Dust Clears," *IEEE Trans. on Knowledge and Data Engineering*, 2(1) (March 1990), pp 143-159.

Harder, T., H. Schoning, and A. Sikeler, "Parallel Query Evaluation: A New Approach to Complex Object Processing," *IEEE Database Engineering, Vol 8* (1989), pp 21-29.

Harmon, S., "The Image Makers," *Workstation News*, 2(2) (February 1991), p 22-23.

Johnson, L.E. et al., "Geographic Information Systems for Hydrologic Modeling," *Proc. 3rd ASCE Water Resources Operations Management Workshop: Computerized Decision Support Systems for Water Managers*, (June 1988), pp 736-749.

Joseph, J.V., et al., "Object Oriented Databases: Design and Implementation," *Proceedings of the IEEE*, 79(1) (Institute of Electrical and Electronics Engineers [IEEE], January 1991), pp 42-64.

Kameny, I., "Global Information System Issues," *Proc.* 5^{th} *Int. IEEE Conf. Data Engineering* (February 1987), pp 672-673.

Ketabchi, M.A., S. Mathur, T. Risch, and J. Chen, "Comparative Analysis of RDBMS and OODBMS: A Case Study," *Proc. IEEE COMPCON 90* (February 1990), pp 528-535.

Kim, W., et al., "Architecture of the ORION Next-Generation Database System," *IEEE Trans. on Knowledge and Data Engineering, 2(1)* (March 1990), pp 109-124.

Lewis, T.G., "Issues in Parallel Programming: Why Aren't We Having Fun Yet?" *Supercomputing Review* (July 1990).

Meador, W.E., "Disk Array Systems," *Proc. IEEE COMPCON 89* (February 1989), pp 143-146.

Miller, R., and Q.E. Stout, "An Introduction to the Portable Parallel Programming Language: Seymour," *Proc. IEEE COMPSAC 89* (September 1989), pp 94-101.

Open Software Foundation, *Application Environment Specification* (Prentice-Hall, 1990).

Page, I., ed., *Parallel Architectures and Computer Vision* (Clarendon Press, 1988).

Patterson, D.A., et al., "Introduction to Redundant Arrays of Inexpensive Disks (RAID)," *Proc. IEEE COMPCON 89* (February 1989), pp 112-117.

Raper, J.F., and M.S. Bundock, "UGIX: A GIS Independent User Interface Environment," *Auto-Carto 10* (March 1991), pp 275-295.

Rasure, J., S. Hallett, and R. Jordan, "A Comprehensive Software System for Image Processing and Programming," *SPIE Proceedings Vol. 1075, Digital Image Processing Applications* (January 1989), pp 37-45.

Rice, T.A., and L.H. Jamieson, "Parallel Processing for Computer Vision" in S. Levialdi, ed., *Integrated Technology for Parallel Image Processing* (Academic Press, 1985).

Rowe, L., and M. Stonebraker, "The POSTGRES Model," *Proc. 12th Very Large Database Conference* (1987), pp 83-96.

Samet, H., *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS* (Addison-Wesley, Reading, MA, 1990).

Schek, H.J., et al., "The DASDBS Project: Objectives, Experiences, and Future Prospects," *IEEE Trans. on Knowledge and Data Engineering* (March 1990), pp 25-43.

Siegel, H.J, et al., "PASM: A Partitionable SIMD/MIMD System for Image Processing and Pattern Recognition," *IEEE Trans. on Computers,* C-30(12) (December 1981), pp 934-947.

Stonebraker, M., "Inclusion of New Types in Relational Database Systems," *Proc. 2nd Int. IEEE Conf. Data Engineering* (February 1986).

Stonebraker, M., and J. Dozier, *Sequoia 2000: Large Capacity Object Servers to Support Global Change Research,* Technical Proposal (University of California, May 1991).

Stonebraker, M., and L. Rowe, "The Design of POSTGRES," *IEEE Database Engineering,* Vol 6 (1987).

Stonebraker, M., et al., "Extensibility in POSTGRES," *IEEE Database Engineering,* Vol 6 (1987).

Stonebraker, M., L. Rowe, and M. Hirohama, "The Implementation of POSTGRES," *IEEE Trans. on Knowledge and Data Engineering,* 2(1) (March 1990).

Stonebraker, M., "Future Trends in Data Base Systems," *IEEE Trans. on Knowledge and Data Engineering,* 1(1) (March 1989), pp 33-44.

Stonebraker, M., et al., "The Design and Implementation of INGRES," *ACM Trans. on Database Systems,* 1(3) (September 1976), pp 189-222.

Stonebraker, M., "The POSTGRES Storage System," *Proc. 12th Very Large Database Conference* (1987).

Warnick, L., and D. Blaylock, "Theory of Revolution," *Workstation News,* 2(2) (February 1991), pp 26-27.

Westervelt, J., "The Two Classes of GIS Users (Or How to Make Software Salad)", *GIS World,* 3(5) (October 1990), pp 111-112.

Wilkes, W., P. Klahold, and G. Schlageter, "Complex and Composite Objects in CAD/CAM Databases," *Proc. 5th Int. Conf. Data Engineering* (February 1989), pp 443-450.

Wilkinson, K., P. Lyngbæk, and W. Hasan, "The IRIS Architecture and Implementation," *IEEE Trans. on Knowledge and Database Engineering,* 2(1) (March 1990), pp 63-75.

Wilms, P.F., P.M. Schwarz, H.J. Schek, and L.M. Haas, "Incorporating Data Types in an Extensible Database Architecture," *Proc. 3rd Int. Conf. Data and Knowledge Bases: Improving Usability and Responsiveness* (June 1988), pp 180-192.

Wu, X., and J.D. Westervelt, *Engineering Applications of Neural Computing: A State of the Art Survey*, Special Report N-91/22/ADA237628 (U.S. Army Construction Engineering Research Laboratory [USACERL], May 1991).

Zimmerman, H., "OSI Reference Model—The ISO Model of Architecture for Open Systems Interconnection," *IEEE Trans. on Communications*, Com-28(4) (April 1980), pp 425-432.

Uncited

Adams, D.A., "Parallel Processing Implications for Management Scientists," *Interfaces* 20(3) (May-June 1990), pp 88-98.

Agrawal, R., N.H. Gehani, and J. Srinivasan, "OdeView: The Graphical Interface to Ode," *Proc. 1990 ACM SIGMOD* (June 1990), pp 34-43.

Ahad, R., "A Methodology for Design and Documentation of Persistent Object Bases of Information Systems," *Proc. IEEE COMPCON 90* (February 1990), pp 333-339.

Andrews, T., and C. Harris, "Combining Language and Database Advances in an Object-Oriented Development Environment," *Proc. OOPSLA '87* (October 1987), pp 430-440.

Ansoult, M.M., P.J. Soille, and J.A. Loodts, "Mathematical Morphology: A Tool for Automated GIS Data Acquisition from Scanned Thematic Maps," *Photogrammetric Engineering and Remote Sensing*, 56(9) (September 1990), pp 1263-1271.

Aoe, J-I., "An Efficient Algorithm of Compressing Decimal Notations for Tree Structures", *Proc. IEEE COMPSAC 89* (September 1989), pp 316-323.

Argialas, D.P., and C.A. Harlow, "Computational Image Interpretation Models: An Overview and Perspective," *Photogrammetric Engineering and Remote Sensing*, 56(6) (June 1990), pp 871-886.

Arnold, J.G., "Simulation of Complex Hydrologic Basins," *Proc. SCS Summer Computer Simulation Conference* (July 1989), pp 682-687.

Aronson, P, "Attribute Handling for Geographic Information Systems," *Auto-Carto 8* (April 1987), pp 346-355.

Bachman, C.W., "A Personal Chronicle: Creating Better Information Systems, with Some Guiding Principles," *IEEE Trans. on Knowledge and Data Engineering*, 1(1) (March 1989), pp 17-32.

Baraniecki, A. Z., "Concurrent Architectures for Real-Time Signal Processing," *Microprocessing and Microprogramming*, 28 (1989), pp 233-238.

Bassiouni, M.A., and C.A. Water, "Efficient Coding for Integrated Information Systems and Multimedia Databases," *Proc. IEEE COMPSAC 89* (September 1989), pp 308-315.

Batory, D.S., T.Y. Leung, and T.E. Wise, "Implementation Concepts for an Extensible Data Model and Data Language," *ACM Trans. on Database Systems*, 13(3) (September 1988), pp 231-262.

Batten, L.G., "National Capital Urban Planning Project: Development of a Three-Dimensional GIS Model," *Auto-Carto 9* (April 1989), pp 336-340.

Batz, T., P. Baumann, and D. Kohler, "Data Model Supporting System Engineering," *Proc. IEEE COMPSAC 88* (October 1988), pp 178-185.

Bennis, K., et al., "Geograph: A Topological Storage Model for Extensible GIS," *Auto-Carto 10* (March 1991), pp 349- 367.

Berry, J.K., "Computer-Assisted Map Analysis: Potential and Pitfalls," *Photogrammetric Engineering and Remote Sensing*, 53(10) (October 1987), pp 1405-1410.

Bestul, T., "A General Technique for Creating SIMD Algorithms on Parallel Pointer-based Quadtrees," *Auto-Carto 10* (March 1991), pp 428-444.

Bitton, D., "Technology Trends in Mass-Storage Systems, Panel," *Proc. 1990 ACM SIGMOD* (June 1990), pp 7-8.

Blaha, M.R., W.J. Premerlani, and J.E. Rumbauch, "Relational Database Design Using an Object-Oriented Methodology," *Communications of the ACM* 31(4) (April 1988), pp 414-427.

Bloom, T., and S.B. Zdonik, "Issues in the Design of Object-Oriented Database Programming Languages," *Proc. OOPSLA '87* (October 1987), pp 441-451.

Bourbakis, N.G., and A. Klinger, "A Hierarchical Coding of Reduced Picture Information," *Proc. IEEE COMPSAC 88* (October 1988), pp 43-48.

Breen, D.E., P.H. Getto, and A.A. Apodaca, "Object-Oriented Programming in a Conventional Programming Environment," *Proc. IEEE COMPSAC 89* (September 1989), pp 334-343.

Bronson, E.C., T.L. Casavant, and L.H. Jamieson, "Experimental Application-Driven Architecture Analysis of an SIMD/MIMD Parallel Processing System," *Proc. 1989 Int. Conf. Parallel Processing* (August 1989), pp I-59-67.

Bruegger, B.P., and A.U. Frank, "Hierarchies Over Topological Data Structures," *Proc. ASPRS-ACSM* (March 1989), pp 137-145.

Bundock, M.S., "An Integrated DBMS Approach to Geographical Information Systems," *Auto-Carto 8* (April 1987), pp 292-301.

Campbell, W.J., and R.F. Cromp, "Evolution of an Intelligent Information Fusion System," *Photogrammetric Engineering and Remote Sensing*, 56(6) (June 1990), pp 867-870.

Caplinger, M., "An Information System Based on Distributed Objects," *Proc. OOPSLA '87* (October 1987), pp 126-137.

Cavano, J.P., "Software Issues Facing Parallel Architectures," *Proc. IEEE COMPSAC 88* (October 1988), pp 300-301.

Chang, S.-K., "Visual Languages: A Tutorial and Survey," *Visualization in Programming: 5th Interdisciplinary Workshop in Informatics and Psychology* (May 1986), pp 1-23.

Chang, S.-K., et al., "An Intelligent Image Database System," *IEEE Trans. on Software Engineering* 14(5) (May 1988), pp 681-688.

Chou, H-C., "Design of a Language for Spatial Analysis," *Proc. ACSM-ASPRS, Vol. 2.* (March 1991), pp 56-64.

Cornish, R.E., "Geographic Databases," *Proc. 2nd Conf. Computing in Civil Engineering* (June 1980), pp 660-669.

Cowen, D.J., "GIS versus CAD versus DBMS: What Are the Differences?" *Photogrammetric Engineering and Remote Sensing*, 54(11) (November 1988), pp 1551-1555.

Cruz, A.J.O., "Parallel Algorithms for SIMD Computers," *Microprocessing and Microprogramming*, 28 (1989), pp 85-90.

Croswell, P.L., and S.R. Clark, "Trends in Automated Mapping and Geographic Information System Hardware," *Photogrammetric Engineering and Remote Sensing*, 54(11) (November 1988), pp 1571-1576.

Delgado-Frias, J.G., and W.R. Moore, eds., *VLSI for Artificial Intelligence*. Boston: Kluwer Academic Publishers (1989).

Desrochers, G.R., *Principles of Parallel and Multiprocessing*, New York: McGraw-Hill (1987).

Devijver, P.A., and J. Kittler, eds., *Pattern Recognition Theory and Applications*. New York: Springer-Verlag (1987).

Dorfman, E., "Architecture of a Large Object-Oriented Database for Remotely Sensed Data," *Proc. ACSM-ASPRS, Vol. 3*. (March 1991), pp 129-143.

Duis, D., and J. Johnson, "Improving User-Interface Responsiveness Despite Performance Limitations," *Proc. IEEE COMPCON 90* (February 1990), pp 380-386.

Dyer, D.S., "apE: Providing Visualization Tools for a Statewide Supercomputing Network," *Proc. 24th Semi-Annual Cray Users Group Meeting* (September 1989), pp 237-241.

Edwards, A.D.N., "Visual Programming Languages: The Next Generation?" *ACM SIGPLAN Notices*, 23(4) (April 1988), pp 43-50.

Egenhofer, M.J., "Designing a User Interface for a Spatial Information System," *Proc. ACSM-ASPRS* (March 1988), pp 149-161.

Egenhofer, M.J., and A.U. Frank, "Object-Oriented Modeling in GIS: Inheritance and Propagation," *Auto-Carto 9* (April 1989), pp 588-598.

Ehlers, M., G. Edwards, and Y. Bedard, "Integration of Remote Sensing with Geographic Information Systems: A Necessary Evolution," *Photogrammetric Engineering and Remote Sensing*, 55(11) (November 1989), pp 1619-1627.

Erbe, R., and N. Sudkamp, "An Application Program Interface for a Complex Object Database," *Proc. 3rd Int. Conf. Data and Knowledge Bases: Improving Usability and Responsiveness* (June 1988), pp 211-226.

Eveleigh, T.J., and K.D. Potter, "Spectral/Spatial Exploitation of Digital Raster Graphic Map Products for Improved Data Extraction," *Auto-Carto 9* (April 1989), pp 348-356.

Fahlman, S.E., "Parallel Processing in Artificial Intelligence," *Parallel Computation and Computers for Artificial Intelligence*, J.S. Kowalik, ed., (Boston: Kluwer 1988), pp 3-19.

Faloutsos, C., T. Sellis, and N. Roussopoulos., "Analysis of Object Oriented Spatial Access Methods," *Proc. 1987 ACM SIGMOD* (May 1987), pp 426-439.

Faust, N.L., "Display Technologies for Remote Sensing/GIS," *Proc ACSM-ASPRS*, Vol. 4 (March 1991), pp 66-76.

Feuchtwanger, M., "Geographic Logical Database Model Requirements," *Auto-Carto 9* (April 1989), pp 599-609.

Fox, G.C., "Issues in Software Development for Concurrent Computers," *Proc. IEEE COMPSAC 88* (October 1988), pp 302-305.

Franklin, W.R., et al., "Uniform Grids: A Technique for Intersection detection on Serial and Parallel Machines," *Auto-Carto 9* (April 1989), pp 100-109.

Freeman, H., ed., *Machine Vision: Algorithms, Architectures, and Systems*. San Diego: Academic Press (1988).

Freeston, M., "The BANG File: A New Kind of Grid File," *Proc. 1987 ACM SIGMOD* (May 1987), pp 260-269.

Ganti, M., et al., "An Object-Oriented Application Development Environment," *Proc. IEEE COMPCON 90* (February 1990), pp 348-355.

Gelsema, E.S., and L.N. Kanal, eds., *Machine Intelligence and Pattern Recognition, Volume 7: Pattern Recognition and Artificial Intelligence*. Amsterdam: Elsevier Science Publishers (1988).

Glinert, E.P., "Towards Software Metrics for Visual Programming," *Int. J. of Man-Machine Studies*, 30(4) (April 1989), pp 425-445.

Gonzalez, R. C., and P. Wintz, *Digital Image Processing*. Reading, Massachusetts: Addison-Wesley (1977).

Graefe, G., "Parallelizing the Volcano Query Processor," *Proc. IEEE COMPCON 90* (February 1990), pp 490-493.

Greenlee, D.D., "Raster and Vector Processing for Scanned Linework," *Photogrammetric Engineering and Remote Sensing*, 53(10) (October 1987), pp 1383-1387.

Gunther, O., "The Design of the Cell Tree: An Object-Oriented Index Structure for Geometric Databases," *Proc. 5th Int. IEEE Conf. Data Engineering* (February 1989), pp 598-605.

Guptill, S.C., "Desirable Characteristics of a Spatial Data Base Management System," *Auto-Carto 8* (April 1987), pp 278-281.

Guttman, A., "R-Trees: A Dynamic Index Structure for Spatial Searching," *Proc. 1984 ACM SIGMOD* (June 1984), pp 47-57.

Haderle, D.J., "Parallelism with IBM's Relational Database2 (DB2)," *Proc. IEEE COMPCON 90* (February 1990), pp 488-489.

Hadipriono, F.C., et al., "The Development of a Knowledge-Based Expert System for Analysis of Drainage Patterns," *Photogrammetric Engineering and Remote Sensing*, 56(6) (June 1990), pp 905-909.

Haeberli, P.E., "ConMan: A Visual Programming Language for Interactive Graphics," *Computer Graphics*, 22(4) (August 1988), pp 103-111.

Hahn, M.A., D.R. Oestreicher, and R.J. Stevenson, "The Evans & Sutherland View of Tomorrow's Supercomputing Environment," *Proc. IEEE COMPCON 89* (February 1989), pp 300-303.

Healy, R.G., and G.B. Desa, "Transputer-Based Parallel Processing for GIS Analysis: Problems and Potentialities," *Auto-Carto 9* (April 1989), pp 90-99.

Hedelman, H., "A Data Flow Approach to Procedural Modeling," *IEEE Computer Graphics and Applications*, 4(1) (January 1984), pp 16-26.

Hepner, G.F., et al., "Artificial Neural Network Classification Using a Minimal Training Set: Comparison to Conventional Supervised Classification," *Photogrammetric Engineering and Remote Sensing*, 56(4) (April 1990), pp 469-473.

Herring, J.R., "Tigris: Topologically Integrated Geographic Information System," *Auto-Carto 8* (April 1987), pp 282- 291.

Herring, J.R., "A Fully Integrated Geographic Information System," *Auto-Carto 9* (April 1989), pp 828-837.

Heytens, M.L., and R.S. Nikhil, "GESTALT: An Expressive Database Programming System," *ACM SIGMOD Record*, 18(1) (March 1989), pp 54-67.

Hsiao, D.K., and M.N. Kamel, "Heterogeneous Databases: Proliferations, Issues, and Solutions," *IEEE Trans. on Knowledge and Data Engineering*, 1(1) (March 1989), pp 45-62.

Ichikawa, T., and M. Hirakawa, "Visual Programming—Toward Realization of User-Friendly Programming Environments," *IEEE Proc. 1987 Fall Joint Computer Conference* (October 1987), pp 129-137.

Illert, A., "Automatic Digitization of Large Scale Maps," *Auto-Carto 10* (March 1991), pp 113-122.

Ingalls, D., et al., "Fabrik: A Visual Programming Environment," *Proc. OOPSLA '88* (September 1988), pp 176-190.

Ingram, K.J., and W.W. Phillips, "Geographic Information Processing Using a SQL-Based Query Language," *Auto-Carto 8* (April 1987), pp 326-335.

Ishikawa, H., "An Object-Oriented Knowledge Base Approach to a Next Generation of Hypermedia System," *Proc. IEEE COMPCON 90* (February 1990), pp 520-527.

Jellinghaus, R., "Eiffel Linda: An Object Oriented Linda Dialect," *ACM SIGPLAN Notices*, 25(12) (December 1990), pp 70-84.

Johnson, L.E., "MAPHYD-A Digital Map-Based Hydrologic Modeling System," *Photogrammetric Engineering and Remote Sensing*, 55(6) (June 1989), pp 911-917.

Johnston, K.M., "Natural Resource Modeling in the Geographic Information System Environment," *Photogrammetric Engineering and Remote Sensing*, 53(10) (October 1987), pp 1411-1415.

Johnston, K.M., et al., "Orpheus: An Integration," *Proc. ACSM-ASPRS* (March 1988), pp 11-22.

Jones, C.V., "An Introduction to Graph-Based Modeling Systems, Pert I: Overview," *ORSA J. on Computing*, 2(2) (Spring 1990), pp 136-151.

Joseph, T., and A.F. Cardenas, "PICQUERY: A High Level Query Language for Pictorial Database Management," *IEEE Trans. on Software Engineering*, 14(5) (May 1988), pp 630-638.

Kasturi, R., and J. Alemany, "Information Extraction from Images of Paper-Based Maps," *IEEE Trans. on Software Engineering*, 14(5) (May 1988), pp 671-675.

Kavouras, M., and S.E. Masry, "An Information System for Geosciences: Design Considerations," *Auto-Carto 8* (April 1987), pp 336-345.

Keating, T., W. Phillips, and K. Ingram, "An Integrated Topologic Database Design for Geographic Information Systems," *Photogrammetric Engineering and Remote Sensing*, 53(10) (October 1987), pp 1399-1402.

Kerckhoffs, E.J.H., "Mini-Supercomputers: New Perspectives in Scientific Computation and Simulation," *Systems Analysis and Simulation I: Theory and Foundations*, A. Sydow, S.G. Tzafestas and R. Vichnevetsky, eds. (Springer-Verlag, Berlin, 1988).

Kim, W., et al., "Features of the ORION Object-Oriented Database System," *Object-Oriented Concepts, Databases, and Applications*, W. Kim and F. H. Lochovsky, eds. (Addison-Wesley, Reading, MA, 1987), pp 251-282.

Kim, W., J. Banerjee, and H.-T. Chou, "Composite Object Support in an Object-Oriented Database System," *Proc. OOPSLA '87* (October 1987), pp 118-125.

Kim, W., et al., "Integrating an Object-Oriented Programming System with a Database System," *Proc. OOPSLA '88* (September 1988), pp 142-152.

Kim, W., "Architectural Issues in Object-Oriented Databases," *J. of Object-Oriented Programming*, 2(6) (March-April 1990), pp 29-38.

King, R., "Semantic and Object-Oriented Database Support for Software Environments," *Proc. IEEE COMPSAC 88* (October 1988), pp 441-442.

Kitiawa, H., and N. Ohbo, "Design Data Modeling with Versioned Conceptual Configuration," *Proc. IEEE COMPSAC 89* (September 1989), pp 225-233.

Kittler, J., and M.J.B. Duff, *Image Processing Architectures* (Research Studies Press, New York, 1985).

Kuhn, W., "Are Displays Maps or Views," *Auto-Carto 10* (March 1991), pp 261- 274.

Lam, H., S.Y.W. Su, and A.M. Alashqur, "Integrating the Concepts and Techniques of Semantic Modeling and the Object-Oriented Paradigm", *Proc. IEEE COMPSAC 89* (September 1989), pp 209-217.

Langran, G., "Producing Answers to Spatial Questions," *Auto-Carto 10* (March 1991), pp 133-147.

Leler, W., "PIX, the latest NeWS," *Proc. IEEE COMPCON 89* (February 1989), pp 239-242.

Leler, W., "Linda Meets UNIX", *IEEE Computer*, 23(2) (February 1990), pp 43-54.

Lin, W., T-L. Sheu, and C.R. Das, "A Quadtree Communication Structure for Fast Data Searching and Distribution," *Proc. IEEE COMPSAC 88* (October 1988), pp 316-323.

Logan, T.L., and N.A. Bryant, "Spatial Data Software Integration: Merging CAD/CAM/Mapping with GIS and Image Processing," *Photogrammetric Engineering and Remote Sensing*, 53(10) (October 1987), pp 1391-1395.

Logan, T.L., et al., "CAD/CAM/Mapping Interfaces with the VICAR/IBIS Image Processing and GIS System," *Proc. ACSM-ASPRS* (March 1987), pp 130-140.

Lucco, S.E., "Parallel Programming in a Virtual Object Space," *Proc. OOPSLA '87* (October 1987), pp 26-34.

Lukatela, H., "GIS Future: Automated Cartography or Geo-Relational Solid Modelling?" *Auto-Carto 9* (April 1989), pp 341-347.

Lupien, A.E., W.H. Moreland, and J. Dangermond, "Network Analysis in Geographic Information Syst ems," *Photogrammetric Engineering and Remote Sensing*, 53(10) (October 1987), pp 1417-1421.

Maeda, A., and J. Shibayama, "Application of Automatic Drawing Reader for the Utility Management System," *Proc. IEEE Workshop on Computer Architecture for Pattern Analysis and Image Database Management* (November 1985), pp 139-145.

Maffini G., "Raster versus Vector Data Encoding and Handling: A Commentary," *Photogrammetric En gineering and Remote Sensing*, 53(10) (October 1987), pp 1397-1398.

Maguire, D.J., B. Kimber, and J. Chick, "Integrated GIS: The Importance of Raster," *Proc. ACSM-ASPRS, Vol. 4* (March 1991), pp 107-116.

Manola, F., J. Orenstein, and U. Dayal, "Geographic Information Processing in the PROBE Database System," *Auto-Carto 8* (April 1987), pp 316-325.

Mark, D.M., and A.U. Frank, "Concepts of Space and Spatial Language," *Auto-Carto 9* (April 1989), pp 538-556.

McAuliffe, K.P., and P.M. Kogge, eds., *Proc. 1989 Int. Conf. Parallel Processing, Volume 1: Architecture* (Penn. State Press, University Park, Pennsylvania 1989).

McCuster, D.E., "Evolution of Large Format Rester Scanning at Defense Mapping Agency," *Proc. ACSM-ASPRS, Vol 4.* (March 1991), pp 117-122.

McDermott, M.H., "The Spatial Data Transfer Standard," *Proc. ACSM-ASPRS, Vol. 2* (March 1991), pp 222-230.

McDonald, J.A., "Painting Multiple Views of Complex Objects," *Proc. ECOOP/OOPSLA '90* (October 1990), pp 245-257.

McDonald, K.R., and D.J. Blake, "Information Management Challenges of the EOS Data and Information System," *Proc. ACSM-ASPRS, Vol 3* (March 1991), pp 258-267.

McLaren, R.A., "Visualization Techniques and Applications within GIS," *Auto-Carto 9* (April 1989), pp 5-14.

Mehldau, G., and R.A. Schowengerdt, "A C-Extension for Rule-Based Image Classification Systems," *Photogrammetric Engineering and Remote Sensing*, 56(6) (June 1990), pp 897-892.

Menon, S., P. Gao, and T. Smith, "Multi-Colored Quadtrees for GIS: Exploiting Bit-Parallelism for Rapid Boolean Overlay," *Pattern Recognition Letters*, 8(3) (October 1988), pp 171-179.

Menon, S., and T.R. Smith, "A Declarative Spatial Query Processor for Geographic Information Systems," *Photogrammetric Engineering and Remote Sensing*, 55(11) (November 1989), pp 1593-1600.

Merrow, T., and J. Laursen, "A Pragmatic System for Shared Persistent Objects," *Proc. OOPSLA '87* (October 1987), pp 103-110.

Mikkilineni, R.V., "Potential Use of the Object Paradigm for Software Engineering Environments in the 1990s," *Proc. IEEE COMPSAC 88* (October 1988), pp 439-440.

Mitter, J., "Tigris Mapper Viewed as a Digital Data Capturing Tool in Object-Oriented Environment," *Auto-Carto 9* (April 1989), pp 367-376.

Mohan, L., and R.L. Kashyap, "An Object-Oriented Knowledge Representation for Spatial Information," *IEEE Trans. on Software Engineering*, 14(5) (May 1988), pp 675-681.

Moller-Jensen, L., "Knowledge-Based Classification of an Urban Area Using Texture and Context Information in Landsat-TM Imagery," *Photogrammetric Engineering and Remote Sensing*, 56(6) (June 1990), pp 899-904.

Moore, D.J., et al., "Vishnu: An Object-Oriented Database Management System Supporting Software Engineering," *Proc. IEEE COMPSAC 88* (October 1988), pp 186-192.

Motro, A., "A Trio of Database User Interfaces for Handling Vague Retrieval Requests," *IEEE Database Engineering*, Vol. 8 (1989), pp 118-125.

Motro, A., "FLEX: A Tolerant and Cooperative User Interface to Databases," *IEEE Trans. on Knowledge and Data Engineering*, 2(2) (June 1990), pp 231-246.

Nadas, T., and A. Fournier, "GRAPE: An Environment to Build Display Processes," *Computer Graphics*, 21(4) (July 1987), pp 75-84.

Newell, R.G., M. Easterfeil, and D.G. Theriault, "Integration of Spatial Objects in a GIS," *Auto-Carto 10* (March 1991), pp 408-415.

Ng, S., "Some Design Issues of Disk Arrays," *Proc. IEEE COMPCON 89* (February 1989), pp 137-142.

Nievergelt, J., H. Hinterberger, and K.C. Sevcik, "The Grid File: An Adaptable, Multikey File Structure," *ACM Trans. on Database Systems*, 9(1) (March 1984), pp 38-71.

Ooi, B.C., R. Sacks-Davis, and K.J. McDonell, "Extending a DBMS for Geographic Applications," *Proc. 5th Int. IEEE Conf. Data Engineering* (February 1989), pp 590-597.

Oren, T.I., "Basis for Advances Simulation: Paradigms for the Future," *Modelling and Simulation Methodology*, M.S. Elzas, T.I. Oren and B.P. Zeigler, eds. (North-Holland, Amsterdam 1989).

Orenstein, J., "Spatial Query Processing in an Object-Oriented Database System," *Proc. 1986 ACM SIGMOD* (May 1986), pp 326-336.

Orenstein, J., and F.A. Manola, "PROBE Spatial Data Modeling and Query Processing in an Image Processing Database Application," *IEEE Trans. on Software Engineering*, 14(5) (May 1988), pp 611-629.

Orenstein J., "Redundancy in Spatial Databases," *Proc. 1990 ACM SIGMOD* (June 1990), pp 294-305.

Orenstein J., "A Comparison of Spatial Query Processing Techniques for Native and Parameter Spaces," *Proc. 1990 ACM SIGMOD* (June 1990), pp 343-352.

Parker, H.D., "GIS Software 1989: A Survey and Commentary," *Photogrammetric Engineering and Remote Sensing*, 55(11) (November 1989), pp 1589-1591.

Pearsall, R.J., and C. Winslett, "GIS the Visualization of Tommorrow," *Proc. ACSM-ASPRS, Vol. 2* (March 1991), pp 333- 335.

Penney, D.J., and J. Stein, "Class Modification in the GemStone Object-Oriented DBMS," *Proc. OOPSLA '87* (October 1987), pp 111-117.

Peterson, J.K., and W. Kuhn, "Defining GIS Data Structures by Sketching Examples," *Proc. ACSM-ASPRS, Vol. 2.* (March 1991), pp 261-269.

Pigot, S., "Topological Models for 3D Spatial Information Systems," *Auto-Carto 10* (March 1991), pp 368-392.

Polychronopoulus, C. D., *Parallel Programming and Compilers* (Kluwer Academic Publishers, Boston 1988).

Potsmesil, M., and F.M. Hoffert, "FRAMES: Software Tools for Modeling, Rendering, and Animation of 3D Scenes," *Computer Graphics*, 21(4) (July 1987), pp 84-93.

Qadah, G.Z., "Database Machines: A Survey", *AFIPS Conference Proceedings*, 1985 National Computer Conference, 54, pp 211-223.

Rado, B.Q., A.S. Bury, and C.C. Smith, "Raster-Vector Integration: Real-World Solutions," *Proc. ACSM-ASPRS, Vol. 4.* (March 1991), pp 166-172.

Ramirez, J.R., "Computer-Aided Mapping Systems: The Next Generation," *Photogrammetric Engineering and Remote Sensing*, 57(1) (January 1991), pp 85-88.

Ramirez, J.R., "Understanding Universal Exchange Formats," *Photogrammetric Engineering and Remote Sensing*, 57(1) (January 1991), pp 89-92.

Rebordao, J.M., and M.C. Proenca, "Multi-Temporal Classification of Remote Sensed Data in Image Processors with a Pipeline Architecture," *SPIE Proceedings Vol. 1075, Digital Image Processing Applications* (January 1989), pp 74-89.

Reiss, S.P., "Visual Languages and the GARDEN System," *Visualization in Programming: 5th Interdisciplinary Workshop in Informatics and Psychology* (May 1986), pp 178-198.

Riggins, R.E., W.D. Goran, and L.J. Schmitt, "Tools for Managing Army Training Lands," *Watershed Management in the Eighties*, E.B. Jones and T.J. Ward, eds. (1985), pp 56-60.

Robinson, J.T., "The K-D-B Tree: A Search Structure for Large Multidimensional Dynamic Indexes," *Proc. 1981 ACM SIGMOD* (1981), pp 10-18.

Robinson, V.B., and A.U. Frank, "Expert Systems for Geographic Information Systems," *Photogrammetric Engineering and Remote Sensing*, 53(10) (October 1987), pp 1435-1441.

Roman, G-C., "Formal Specification of Geographic Data Processing Requirements," *Proc. 2th Int. IEEE Conf. Data Engineering* (February 1986), pp 434-446.

Rouse, M.J., and R.F. Cranfield, "An Overview of the Trends and Future Direction of Information Technology in the Water Industry," *Water Science Technology*, 21(10-11) (1989), pp 1129-1135.

Roussopoulos, N., and D. Leifker, "Direct Spatial search on Pictorial Databases Using Packed R-Trees," *Proc. 1985 ACM SIGMOD* (May 1985), pp 17-31.

Roussopoulos, N., C. Faloutsos, and T. Sellis, "An Efficient Pictorial Database System for PSQL," *IEEE Trans. on Software Engineering*, 14(5) (May 1988), pp 639-650.

Roussopoulos, N., L. Mark, T. Sellis, and C. Faloutsos, "An Architecture for High Performance Engineering Information Systems," *IEEE Trans. on Software Engineering*, 17(1) (January 1991), pp 22-33

Rowe, L.A., P. Danzig, and W. Choi, "A Visual Shell Interface to a Database," *Software-Practice and Experience*, 19(6) (June 1989), pp 515-528.

Ryerson, R., "Image Interpretation Concerns for the 1990s and Lessons from the Past," *Photogrammetric Engineering and Remote Sensing*, 55(10) (October 1989), pp 1427-1430.

Salge, F., and M.N. Sclafer, "A Geographic Data Model Based on HBDS Concepts: The IGN Cartographic Data Base Model," *Auto-Carto 9* (April 1989), pp 110-117.

Sellis, T.K., "Efficiently Supporting Procedures in Relational Database Systems," *Proc. 1987 ACM SI GMOD* (May 1987), pp 278-291.

Sellis, T., N. Roussopoulos, and C. Faloutsos, "The R_ Tree: A Dynamic Index for Multidimensional Objects," *Proc. 12th Very Large Database Conference* (September 1987), pp 507-518.

Sellis, T.K. and L. Shapiro, "Query Optimization for Non-traditional Database Applications," *IEEE Trans. on Software Engineering*, 17(1) (January 1991), pp 77-86.

Serra, J., *Image Analysis and Mathematical Morphology* (Academic Press, Orlando 1982).

Shah, A.V., et al., "DSM: An Object-Relationship Modeling Language," *Proc. OOPSLA '89* (October 1989), pp 191-202.

Singh, A., J. Schaeffer, and M. Green, "A Template-Based Approach to the Generation of Distributed Applications Using a Network of Workstations," *IEEE Trans. on Parallel and Distributed Systems*, 2(1) (January 1991), pp 52-67.

Smith, D.R., and A. R. Paradis, "Three-Dimensional GIS for the Earth Sciences," *Auto-Carto 9* (April 1989), pp 324-335.

Smith, K., and S.B. Zdonik, "Intermedia: A Case Study of the Differences Between Relational and Object-Oriented Database Systems," *Proc. OOPSLA '87* (October 1987), pp 452-465.

Soucek, B., and M. Soucek, *Neural and Massively Parallel Computers*, The Sixth Generation. (Wiley, New York 1988).

Srinivasan, R., and S. Mohan, "Integrated System for Neural Network Based Pattern Recognition," *SPIE Proceedings Vol. 1075, Digital Image Processing Applications* (January 1989), pp 328-335.

Strebel, D.E., et al., "The FIFE Information System," *IEEE Trans. on Geoscience and Remote Sensing*, 28(4) (July 1990), pp 703-709.

Su, S.Y.W., "Extensions to the Object-Oriented Paradigm," *Proc. IEEE COMPSAC 89* (September 1989), pp 197-199.

Tanaka, M., and T. Ichikawa, "A Visual User Interface for Map Information Retrieval Based on Semantic Significance," *IEEE Trans. on Software Engineering*, 14(5) (May 1988), pp 666-670.

Tauber, M.J., "On Visual Interfaces and Their Conceptual Analysis," *Visualization in Programming: 5th Interdisciplinary Workshop in Informatics and Psychology* (May 1986), pp 106-123.

Thuraisingham, M.B., "Security in Object-Oriented Database Systems," *J. of Object-Oriented Programming*, 2(6) (March-April 1990), pp 18-25.

Tomlinson, C., and M. Scheevel, "Concurrent Object-Oriented Programming Languages," *Object-Oriented Concepts, Databases, and Applications*, W. Kim and F. H. Lochovsky, eds. (Addison-Wesley, Reading, MA 1987).

Ton, J., J. Sticklen, and A.K. Jain, "Knowledge-Based Segmentation of Landsat Images," *IEEE Trans. on Geoscience and Remote Sensing*, 29(2) (March 1991), pp 222-231.

Turner, A.K., and K.E. Kolm, "Three-Dimensional Geoscientific Information Systems for Groundwater Modeling," *Proc. ACSM-ASPRS, Vol. 4.* (March 1991), pp 217-226.

Tuzhilin, A., and Z.M. Kedem, "Querying and Controlling the Future Behavior of Complex Objects," *Proc. 5th Int. IEEE Conf. Data Engineering* (February 1989), pp 434-442.

Ulgen, O.M., and T. Thomasma, "Computer Simulation Modeling in the Hands of Decision-Makers," *Proc. SCS Western Multiconference, 1989: Simulation and AI* (January 1989), pp 89-95.

Valimohamed, K.A., and L.A. Lopez, *Architecture Independent Environment for Developing Engineering Software on MIMD Computers*, Civil Engineering Systems Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois, Tech. Report UILU-ENG-90-20001 (January 1990).

Van Cleynenbreugel, J., et al., "Delineating Road Structures on Satellite Imagery by a GIS-Guided Technique," *Photogrammetric Engineering and Remote Sensing*, 56(6) (June 1990), pp 893-898.

Van Oosterom, P., "The Reactive-tree: A Storage Structure for a Seamless, Scaleless Geographic Data base," *Auto-Carto 10* (March 1991), pp 393-407.

Ventura, S.J., "Conversion of Automated Geographic Data to Decision-Making Information," *Photogrammetric Engineering and Remote Sensing*, 56(4) (April 1990), pp 511-516.

Verts, W.T., and C.L. Thomson, "Parallel Architectures for Geographic Information Systems," *Proc. ACSM-ASPRS* (March 1988), pp 101-107.

Verts, W.T., "Object-Oriented Spatial Data Structures," *Proc. ACSM-ASPRS, Vol. 3.* (March 1991), pp 455-462.

Wah, B.W., and G-J Li, "A Survey on Special-Purpose Computer Architectures for AI," *Parallel Computation and Computers for Artificial Intelligence*, J.S. Kowalik, ed. (Kluwer, Boston 1988), pp 263-291.

Watanabe, T., Y. Yoshida, and T. Fukumura, "Editing Model Based on the Object-Oriented Approach," *Proc. IEEE COMPSAC 88* (October 1988), pp 67-74.

Weikum, G., "Set-Oriented Disk Access to Large Complex Objects," *Proc. 5th Int. IEEE Conf. Data Engineering* (February 1989), pp 426-433.

Weil, F., L. Jamieson, and E. Delp, "Some Aspects of an Image Understanding Database for an Intelligent Operating System," *Proc. IEEE Workshop on Computer Architecture for Pattern Analysis and Machine Intelligence* (October 1987), pp 203-208.

Weiser, S.P., and F.H. Lochovsky, "OZ+: An Object-Oriented Database System," *Object-Oriented Concepts, Databases, and Applications*, W. Kim and F. H. Lochovsky, eds. (Addison-Wesley, Reading, MA 1987), pp 309-337.

Whang, K.-Y., "A Seamless Integration in Object-Oriented Database Systems," *Proc. 5th Int. IEEE Conf. Data Engineering* (February 1989), pp 675-676.

Wu, C.T., "Benefits of Object Oriented Programming in Implementing Visual Database Interface," *J. of Object-Oriented Programming*, 2(6) (March-April 1990), pp 8-16.

Wu, X., and J.D. Westervelt, *Engineering Applications of Neural Computing: A State of the Art Survey*, Special Report N-91/22/ADA237628, U.S. Army Construction Engineering Research Laboratory [USACERL] (May 1991).

Zdonik, S., "Directions in Object-Oriented Databases", *Proc. IEEE COMPSAC 89* (September 1989) p 200.

Zdonik, S., and D. Maier, eds., *Readings in Object Oriented Database Systems* (Morgan-Kaufmann, San Mateo, CA 1990).

Zeller, H., "Parallel Query Execution in NonStop SQL," *Proc. IEEE COMPCON 90* (February 1990), pp 484-487.

Zhou, Q., "A Method for Integrating Remote Sensing and Geographic Information Systems," *Photogrammetric Engineering and Remote Sensing*, 55(5) (May 1989), pp 591-596.

ABBREVIATIONS AND ACRONYMS

AES	Application Environment Specification
apE	animation production Environment
API	application programming interface
APP	application
DB2	Database 2
DBMS	database management system
DRAM	dynamic random access memory
EN	Environmental Division
ERDAS	Earth Resources Data Analysis System
ESRI	Environmental Systems Research Institute
GIS	geographic information system
GRASS	Geographic Resources Analysis Support System
IEEE	Institute of Electrical and Electronics Engineers
ISO-OSI	International Standards Organization Open Systems Interconnection
MIMD	multiple instruction multiple data
NF^2	non-first normal form (database)
ODE	Object Database and Environment
OODBMS	object-oriented database management system
OSF	Open Software Foundation
PASM	partitionable SIMD/MIMD
PE	processing element
QI	query interface
RDBMS	relational database management system
SIMD	single instruction multiple data

SSD	solid state disk
USACERL	U.S. Army Construction Engineering Research Laboratory
VPE	visual programming environment

USACERL DISTRIBUTION

Chief of Engineers
ATTN: CEHEC-IM-LH (2)
ATTN: CEHEC-IM-LP (2)
ATTN: CERD-L
ATTN: CECC-P
ATTN: CECW
ATTN: CECW-O
ATTN: CECW-P
ATTN: CECW-RR
ATTN: CEMP
ATTN: CEMP-M
ATTN: CEMP-O
ATTN: CEMP-R
ATTN: CEMP-C
ATTN: CEMP-E
ATTN: CERD
ATTN: CERD-C
ATTN: CERD-M
ATTN: CERM
ATTN: DAEN-ZCE
ATTN: DAEN-ZCI
ATTN: DAEN-ZCM
ATTN: DAEN-ZCZ

CEHSC
ATTN: CEHSC-ZC 22060
ATTN: DET III 79906
ATTN: CEHSC-F 22060
ATTN: CEHSC-TT-F 22060

US Army Engineer Districts
ATTN: Library (41)
Alaska 99506
ATTN: NAPEN-PL

US Army Engr Divisions
ATTN: Library (13)

416th Engineer Command 60623
ATTN: Facilities Engineer

US Military Academy 10996
ATTN: Facilities Engineer
ATTN: Dept of Geography & Environmental Engrng
ATTN: MAEN-A

Fort Belvoir, VA 22060
ATTN: Australian Liaison Officer
ATTN: Water Resource Center
ATTN: Engr Strategic Studies Ctr
ATTN: Topographic Engr Center
ATTN: CECC-R

CECRL, ATTN: Library 03755

CEWES, ATTN: Library 39180

Norton AFB, CA 924-0
ATTN: AFRCE-MX/DE

Tyndall AFB, FL 32403
AFESC/Engineering & Service Lab

NAVFAC
ATTN: Division Offices (11)
ATTN: Facilities Engr Cmd (9)
ATTN: Naval Public Works Center (9)
ATTN: Naval Constr Battalion Ctr 93043
ATTN: Naval Civil Engr Lab 93043 (3)

Engineering Societies Library
New York, NY 10017

National Guard Bureau 20310
Installation Division

US Government Printing Office 20401
Receiving/Depository Section (2)

US Army Env. Hygience Agency
ATTN: HSHB-ME 21010

American Public Works Association 60637

Nat'l Institute of Standards & Tech 20899

Defense Technical Info. Center 22304
ATTN: DTIC-FAB (2)

174
11/91