GRADIS - The Strässle Approach to a Modern GIS

Jack Campbell and Franz Steidler

STI Strassle - Technische Informationssysteme
Kanalstrasse 33, CH-8152 Glattbrugg
Telephon: +41-1-8288111 Fax: +41-1-8288212

ABSTRACT

The introduction of Geographic Information Systems (GIS) into organizations is a major investment in data capture, data storage, and data analysis technology, it is a new technology that provides organizations an effective means of accessing, sharing, and analyzing geographic data for decision support. Its implementation is not trivial because a variety of human, organizational, and technological factors must be considered for success. The longevity of data and the interdisciplinary use of data are the most important requirements for a GIS because they directly relate to cost-benefit. Open systems fulfill these important requirements. This paper discusses strässle's concept for a modern GIS and the four architectural principles of their GIS system, GRADIS, that were incorporated into its design to implement this concept.

Index Terms - Geographic Information Systems (GIS), open system architecture, integrated relational database, distributed processing, concurrent database processing, standards, heterogeneous hardware environment, built-in analysis tools, spatial data structures, macro language, Graphical User Interface (GUI), GIS cost-benefit.

INTRODUCTION

The importance of Geographic Information Systems (GIS) is dramatically rising as municipalities, various government services, and environmental agencies strive to improve their capabilities for effective information processing.

This paper outlines the underlying philosophy of strässle's GIS system GRADIS. No attempt has been made to describe the entire system or to examine all the technical aspects of a GIS. Only the details necessary to support the conceptual philosophy for strässle's approach to a modern GIS are presented.

There are many types of systems on the market today which handle various aspects of Geographic Information Systems such as Mapping and CAD systems, business and statistical packages, and so forth. Each of these systems typically handles certain problems favorably and has short comings when applied to other types of problems.
Being in the information age, new or potential users of GIS typically have numerous data from various software packagee and a distinct desire to amalgamate this information into a complete system environment. Their basic problem is to differentiate between the vast amount of software and hardware available on the market today. At first glance, GIS, CAD, and Mapping systems all appear to be similar. All these systems use comparable facilities for geometric processing yet on closer inspection they are distinctly different. A GIS system differs greatly from CAD and Mapping in that it integrates geometric processing with various statistical and analysis features into one homogeneous and transparent information processing system. Analysis of geographically referenced data becomes the most dominant feature of a GIS and is also the most difficult aspect for new users to initially comprehend.

For users to obtain maximum benefit from GIS technology, there are generally four important system requirements to be considered for selection criteria. These are:

1 - the possibility to share data with other users, applications, and external systems
2 - Durable data structures and a durable data model to insure the survival of data over a very long project life cycle
3 - Data consistency, security, and referential integrity
4 - Application-specific funcitionality

To satisfy these criteria, strässle considered the following four architectural principles vital in the original GRADIS design:

1 - An open device independent system architecture based on international standards
2 - A higly structured data model

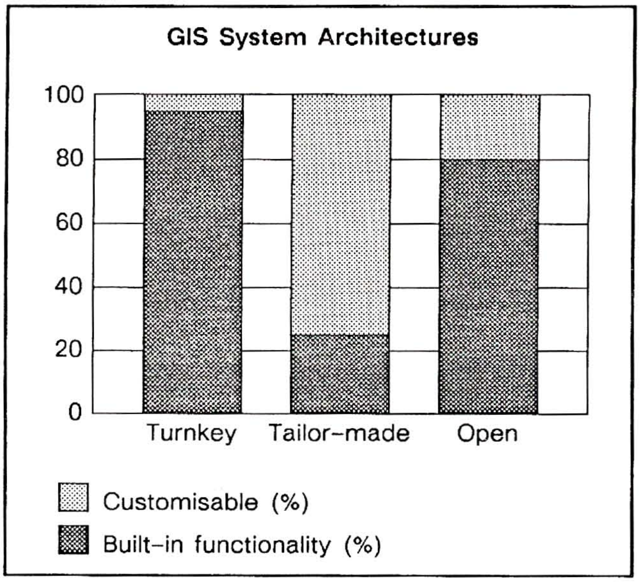
3 - A fully integrated approach whereby all spatial and non- spatial data can be accomodated in the same database system
4 - Built-in GIS functions accessible by an integrated Macro Command Language

To achieve maximum benefit from a GIS investment, potential users have to consider a GIS that incorporates these important architectural principles in the system kernel. Strässle design and developed GRADIS with the permise "think user". The above architectural principles were incorporated into the basic design with the goal of maximising user benefits.

1. STRÄSSLE's PHILOSOPHY

1.1 GIS System Architectures

There are essentially three types of GIS system architectures available in the market place today: 1) "turnkey", 2) "tailor made", and 3) "open". These systems can be classified according to the amount of application-specific functionality that is immediately available to users ("built-in") and the amount of required functionality that must be added with available customisation facilities. We first provide the reader our view of the advantages and disadvantages of these three architectures as a background to support our develoment philosophy for a GIS.



1.1.1 "Turnkey" Systems

The major appeal of "turnkey" systems is that they are "What You See Is What You Get" (WYS/WYG) systems. Systems in this category are touted by their vendors as having all the functionality you need for a reasonable price. They usually can be demonstrated to convince users that their application-specific functionality requirements are met. In other words, you can see you are buying. This is reassuring, especially for first-time buyers, because someone else has done it before (i.e., a vendor has built this system based on their or someone else's specific application requirements). The reality is, though, that no one can design a system for your operations without involving you. The problems with these systems begin when users become experienced and want to customise the system to their work or add additional functionality as their requirements change. In most cases, these systems are difficult or impossible to change because they are built with a pre-determined set of functionality and have no customising tools available for users. If change is necessary, usually the only economical alternative is to replace the system. For these reasons, "turnkey" systems generally have the lowest cost, but have the shortest life cycle.

1.1.2 "Tailor Made" Systems

These systems are normally offered as a set of application programming libraries together with an engineering contract with the global to build you the perfect "tailor made" system for your applications. They have minimum functionality to begin with and are essentially "built from scratch". The good news is that you get a system especially for your needs with a small initial investment. The bad news is that though the costs are relatively minor at the beginning, these costs are growing all the time because of the uncertainties and delays common to all major software development projects. Developing a new and complicated system is a big risk and will probably cost the most over the long term. The message here is don't begin from scratch.

1.1.3 "Open" Systems

Systems in this category usually have 70 to 90 percent of their functionality built in and directly accessible for customising and extending as requirements change. The remaining 10 to 30 percent of the required application functionality is added by users with a set of integrated application-building tools. The term "open" for these systems means open for users to easily customise, enhance, extend, exchange data, and integrate with other applications. The permise is that the users know their application needs best and if they have a set of easy-to-use application tools, they will be able to make the best system for their operations. And since a majority of the functionality is available and accessible, they won't be starting from scratch. Open systems have the advantage of being higly adaptable to specific customer requirements with minimal development risk to meet both current and future needs. Open systems are usually more expensive in the beginning, but the lowest overall cost during the long life cycles of typical GIS projects.

1.2 CAD versus Mapping versus GIS

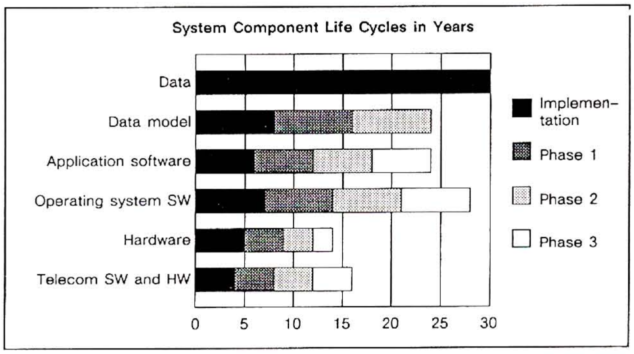
Strässle believes a GIS should not try to compete directly with traditional "turnkey" CAD or Mapping systems. CAD systems provide users powerful graphic functions such as duplicate, mirror, etc. for efficient mechanical and construction design. Mapping systems provide users powerful graphics functions such as COGO (coordinate geometry), interfaces to field data recorders, and plot facilities for efficient map creation, revision, and production. Both of these traditional systems concentrate their functionality on efficient production of technical drawings and maps. While this sort of functionality is important in some cases, it does not solve the problems of geographic information processing. For production efficiency, mapping systems including CAD generally operate on geometry and text primitives that have no logical, topological, or spatial context. These systems are commonly referred to in the literature as "spaghetti" systems because of their lack of intelligent data structures. A GIS typically operates on "object" containing the geographhic representation of a spatial feature combined with directly and indirectlly related logical attribute data. This object data structure is necessary for efficient data analyses. The geometric handling during data capture or data output comprises only a small part of the system.

1.3 An Open System Accomodates Data Sharing

An important task of a GIS is to communicate with other systems to share data, For example, with Ethernet-TCP/IP functionality and data base connects, a user can share data with other users or data can be shared among interdisciplinary applications. Sharing data reduces the costs of data capture, minimizes data redundancy, and promotes effective and consistent data communication between users and departments. An open system architecture with standards allows users to share data effectively within an operational GIS environment.

1.4 A Higly Structured Model Promotes Data Longevity

Another important task of a GIS is to mantain data over the long life cycles common to GIS projects. This can be accomplished with a hierarchy of durable and robust data structures that start at the lowest with objects, have intermediate data structures that contain object coverages designed for long transactions, and at the top, have data structures that contain valid sets of object coverages for archiving. Objects encapsulate the spatial, topological, and logical intelligence of real world geographical entities into one highly-structured data model definition. They have an important advantage of informational context in both a spatial and logical sense. Objects have the potential, when combined with transiction and archiving structures, to be of use for many interdisciplinary applications over long periods of time.



1.5 A Fully Integrated Database is the Key to Data Integrity

When sharing data, users and organizations must have assurance that this data is accessed only by authorized users and applications and is not unknowingly changed simultaneously by different authorized users. To maintain data integrity, mechanism and structures must be provided for data consistency, protection against unauthorized access, and protection against data loss. The preservation of valid GIS data can be most effectively achieved with a fully integrated database scheme whereby all spatial and non-spatial data reside in a single file system.

1.6 A GIS Built for Analysis

Strässle considers a GIS to be a system for data capture/edit, storage, analysis, and output of spatially related data. The analysis part is emphasized because we believe that a significant portion of a GIS must concentrate on the analysis of data to support decision making. For operational efficiency the analysis functions must be built-in to the kernel and directly accessible to users without special programming. This analysis consists of algebraic computations such as polygon analysis (i.e. polygon overlay) and network analysis (i.e. graph tracing). A key functionality of a GIS is to combine attribute processing with polygon analysis and network tracing. These analyses may be very complex - spatial and attribute data may be combined in any order. SQL queries of the geographic database can be performed using attributes as qualifiers. The results of these queries are sets of data that can be reported or displayed thematically as the final analysis objective or can be used as intermediate results for further processing and reporting. The combinations and permutations are endless.

2. THE FOUR ARCHITECTURAL PRINCIPLES

The four architectural principles of GRADIS provide the user a sound basis for assuring that his/her major system requirements are met. Of course, the functionality of a GIS to perform application-specific tasks is important and cannot be overlooked as selection criteria. But if this required functionality resides in a system that does not include all four principles, the user will encounter many maintenance and production problems and will most likely fail to achieve the anticipated cost- benefits of the investment.

2.1 Principle One: An Open Device Independent System Architecture

2.1.1 The Adherence to Standards

We offer our customers an open system with GRADIS. An open system allows users to easily access their data. The data is object- structured but not rigidly bound to any formats or application programs. An open system is hardware independent. Standards are used wherever possible to simplify data interchange and to provide consistent interfaces to applications and foreign data bases. Making a new application simply means defining a new data model and creating the corresponding user interface.

Standards are essential because only they provide the means for the system to be portable, extendable, and integratable over the long GIS project life cycle.

In 1987, Strässle made a major decision to switch the operating system from a VAX based VMS[1] host environment to UNIX[2] together with the decision to no longer use FORTRAN. All source code is being written in the C language. The idea was to make the system truly portable and adaptable to a variety of platforms.

By moving to UNIX, certain vendor specific dependencies are eliminated. Most dialects of UNIX are similar and provide certain defacto standards such as X-11 support and a higher level window manager such as Motif or Open Look. Included is the TCP/IP protocols of Ethernet. This creates a transparent distributed computing environment. With the addition of the eXtended Data Representation (XDR) software originally conceived by SUN, it becomes possible to have a machine hardware independent data transfer offering a high degree of flexibility among various configurations of hardware.

The ORACLE[3] Relational Database Management System forms the nucleus of and is fully integrated with the GRADIS Geographic Information System. This architectural design allows the user to take direct advantage of the present and future capabilities of the ORACLE multi-user, distributed data processing technologies. While database products other than ORACLE may provide similar or more functionality, ORACLE has a large installed customer base and is committed to standards and distributed processing. We consider their potential for longevity to be high because of the wide acceptance of their product by users worldwide - in effect their database is a defacto standard. ORACLE uses the Structured Query Language (SQL) standard. Their commitment to distributed processing technology and standards is also consistent with our architectural principles.

GRADIS considerably simplifies the integration of standards-based third -party software applications with its open system architecture and its adherence to standards. For example, the UNIRAS[4] business graphics package was easily integrated within GRADIS because of the common usage of the Computer Graphic Metafile (CGM) standard.

GIS's can be classified according to their graphics engine as: vector-based, raster-based, or so-called hybrid systems which combine vector and raster storage. Strässle developed GRADIS as a vector based system and easily extended it to a hybrid system, by the addition of a raster

1 VAX and VMS are registered trademarks of Digital Equipment Corporation.

2 UNIX is the registered trademark of Bell Telephone Laboratories, Inc., California, USA.

3 ORACLE is the registered trademark of ORACLE Software Corporation, Belmont, California.

4 UNIRAS is the registered trademark of UNIRAS A/S, Soborg, Denmark.

background module, because of its open architecture. This module provides GRADIS users the capability to overlay and capture vector data on a raster image. The objective of this module is to provide users a simple yet flexible method of super-imposing vector data with raster data for applications that need map data only as an orientation reference. Raster is particularly suited for Utility operations where the focus is on the object data and topology for line and node networks and cadastre is used as a visual location reference. The raster image is registered to a Project data set by a pair of lower left and upper right coordinate pairs. A zooming function allows users to rapidly zoom-in to selected areas with up to 10X magnification depending on image quality. The module includes a raster database for storage and retrieval. A variety of input and output formats are supported. The raster component is based on a package from Industrieanlagen Betriebsgesellschaft GmbH (IABG), Ottobrun and integrated into the graphics engine.

2.1.2 Integration with Existing External Data

In most cases customers begin to work with a GIS already possessing a large amount of data. Customers generally have worked with some form of older technology system before upgrading to a GIS. With such an upgrade comes exisitng data which may or may not already reside in a database.

Data transfer to and from foreign systems is an important consideration. Data may exist in a database and be used along side a GIS in which case the system is not foreign but it's usage is not known by the GIS. In other cases data may be produced from some external application and the objective is to integrate this data into the GIS. For these purposes, GRADIS provides users two data integration tools.

2.1.2.1 Data Interchange Module (DIM)

GRADIS provides a module for data interchange that uses a documented ASCII interchange format. The DIM is an "engine" for loading data from an ASCII file that is structured in a manner dictated by GRADIS. The basic idea is to have a standard data interchange format that is readable as ASCII text and simple to understand and use. The DIM understands GRADIS objects and supports two-way exchange of data. Data interchange formats such as IGES, DFX, SIF and others have no object structure; the data meaning is user dependent and must be interpretated by a filter program and converted to the ASCII DIM format. By providing a generalized data interchange format, end users can create filters with standard UNIX programming tools to exchange data via the DIM.

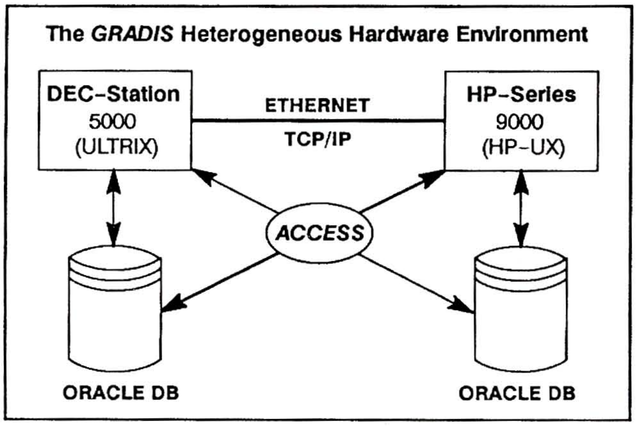
The DIM operates in the batch mode on a GRADIS Project database. A simple and powerful way to integrate exisitng application programs with GRADIS is to modify these programs' output to write the DIM format. The DIM "engine" can then be used to load this output directly to a GRADIS project.

2.1.2.2 Concurrent database Access Module (CDA)

GRADIS provides users both distributed and concurrent access technologies within a network of clients and serves. The Concurrent Database Access module is a mechnaism that allows users to connect external data with GRADIS during an interactive working session. With this module, external multi-user database applications can access and interact with GIS data distributed over a network. The Concurrent Database Access module allows an external SQL*FORMS application to attach to a GRADIS process whereby this application has direct access to both GRADIS object information and external ORACLE information while maintaining data integrity. The objective here is to link spatial objects with existing indirectly related non-spatial data without disturbing the data or the productive use of existing applications. Concurrent Database Access appears as one homogeneous process to a user and is an ideal method of integrating existing database applications with GRADIS. Unlike other systems, concurrent database processing is possible with GRADIS because of its integrated database architectural principle.

2.1.3 Heterogeneous Hardware

Our company strategy is to use international software standards and thereby achieve hardware and software independence. GRADIS is currently available on UNIX workstations from Hewlett-Packard and Digital Equipment Corporation. A heterogeneous mixture of UNIX workstations can be used within the same Ethernet TCP/IP Local Area Network; users can access an ORACLE database located on any workstation or server from any other workstation within the network. Our software was designed to be inherently portable between different versions of UNIX because we have isolated the device and operating system dependencies with our model software architecture. This design considerably simplifies porting from one workstation vendor's operating system to another's.



We recently demonstrated the flexibility of GRADIS to connect to databases installed on different hardware at the Gemeinde '91 Exhibition in Berne, Switzerland. Using an Ethernet-TCP/IP local area network, a HP 9000/400 (HP-UX) was connected to a DECstation 5200 (DEC-ULTRIX) and a VAXstation 3100 (DEC-VMS). A different GRADIS/ORACLE database was installed on all three computers and GRADIS software was installed on the two UNIX workstations. The transparent access to any of the three databases from the two UNIX workstations was demonstrated.

2.2 Principle Two: A Highly Structured data Model

A GIS provides a data base for storing and compilation of any spatially related data. The difference between CAD and GIS is that a GIS allows the reconstruction of existing objects and their relations, whereas a CAD is used to construct new objects. The difference between Mapping and GIS is that the GIS models and analyses the real world, whereas a Mapping system creates and compiles digital maps. CAD and Mapping systems are optimized for drawing and map documentation and their analysis capabilities are limited due to their simple graphic and text data structures. A GIS is designed for analysis and it has intelligent data structures to support this objective.

GRADIS organises geographical data into objects consisting of both geometric (graphic) and logic (attribute) data components. The topological and spatial relationship inherent in the GRADIS data model are encapsulated in object definitions and implemented via ORACLE table structures. Objects are logically grouped for administrative convenience into coverages or Layers. One or more Layers reside in a Project, which is a working subset of the valid database, called the Archive. This data organization combined with the functionality of ORACLE SQL*NET form the basis of a powerful distributed database processing capability for GRADIS.

The GRADIS object definition provides a clear separation of the logical data model from the graphical representation. The benefits here are data consistency, representation flexibility, and productivity. A complete class of objects can be uniformly assigned new graphic representations (i.e., color, line style, fill pattern, or symbols) interactively with a single user action without disturbing the logical integrity of the objects in any way.

The GRADIS data structures for Projects and Objects provide the basis for performing simulations required for planning at a high level of user abstraction:

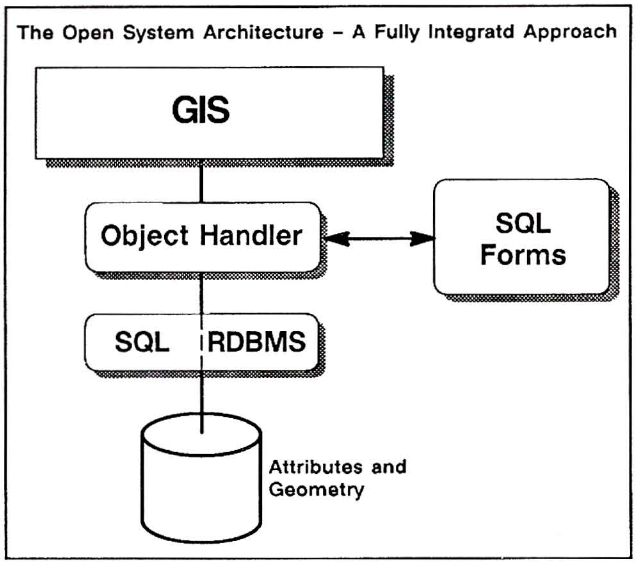
- Temporary data from external sources (i.e., real-time control systems, environmental monitoring systems) can reside in a Project; this data can then be entered into the database or deleted using the three options described above.
- A new object type can be assigned to newly created objects after each stage of the simulation. The user can then find out if the data is from the original or from the simulation by executing a simple SQL*PLUS command.
- Set operations also can be used to store the intermediate results of iterative simulation steps.

Users can access objects (the internal data structure of GRADIS) directly with the GRADIS Command Language. These commands provide users the ability to manipulate objects directly within the defined data model in a natural and intuitive manner.

2.3 Principle Three: A Fully Integrated Approach

The GRADIS database has a fully integrated approach whereby all spatial and non-spatial data can be accomodated in the same database system. A GRADIS database is one physical ORACLE database. This database consists of one or more "archives" representing the top level master data store. Archives each have a self-contained specific GIS data model. Within the archive there exists a hierarchical model defining the "administrative" data and the data definition of the user's data model. All geometric data is included under one roof in ORACLE. There are no external files containing geometry.

Users operating on a working database copy of an Archive known simply as a Project. These Projects are geometrical and logical subsets of the Archive's data and provide data security during data editing and analysis. The provides the system with a mechanism for long transaction manage-



ment, an operational necessity for GIS data integrity. Typically database systems such as ORACLE provide only short transaction management.

GRADIS also provides a data structure for grouping object coverages that is a form of layering. Layers provide a mechanism for organizing data administratively (i.e., according to departmental responsabilities) and for controlling data security. Each layer contains a number of object classes combined with a set of users' access rights to this data. Only users with read- write access can modify data. Additional layers can be accessed simultaneously but can only be viewed or referenced for geometric construction. To facilitate fast access to spatial data, a spatial retrieval and update mechanism is provided based on a modified version of a grid file[1]. Fast access to objects within the relational data base is gained by using the GRID File Mechanism, which is based on a modified QUAD TREE structure. The GRID adjusts dynamically as a function of object density. This spatial access method is not used for direct analysis but as a tool to rapidly load a selected Project's data from a large Archive. Once inside a Project where the data sets can be assumed to be smaller, the spatial access can be handled directly by visually selecting the graphical elements.

The database is the kernel of a GIS. It is the basis for queries and analyses, it connects attribute data to positions and geographic entities and contains the objects and their relations. The GIS database forms a model of the real world. Spatial data is referenced in a topologically structured model. The spatial objects are represented by points, lines and areas and contain the attribute data as well. Spetial relations can be defined among objects within the data model directly and other relationships can be computed with polygon analysis functions as required. The integrated GRADIS database controls access rights, provides recovery mechanisms, long transactions, and many other analysis and management tools. All of this functionality can only be provided and managed effectively with an integrated database architecture.

2.4 Principle Four: Built-in GIS Functions

2.4.1 Application Development Tools

A major objective of Strässle's was to include an extensive set of generic capture, edit, and analysis tools for GIS applications within the GRADIS kernel that are direclty accessible to users via menu selections without special programming. These built-in analysis tools avoid the high costs and development risks of programming normally required of other GIS and Mapping systems to obtain the same level of analysis functionality already available with GRADIS.

GRADIS provides complete functionality for querying multiple types of data. The Graphic Command Language (GCL), ORACLE SQL, and the GRADIS extended SQL can be used individually or in combination to perform an almost endless variety of spatial and a spatial queries on geographic data. The Macro Command Language (MCL) can be used to combine these functions logically under one named command and is described later in more detail. The stored data in GRADIS is seamless (with no map sheet boundary limitations) and queries can be performed on the total database (the Archive) even though a portion of the database (the Project) is being viewed. The hierarchically structured GRADIS Command Language (GCL) has SELECT and ENQUIRE commands which can query objects using both spatial and logical qualifiers. Users have also the possibility to perform graphic and alpha-numeric queries of the database with SQL, using a combination of spatial and attribute qualifiers. The user can perform these queries using ORACLE tools such as SQL*PLUS and SQL*FORMS. The query results can be displayed in graphic windows or directed to files for a variety of formatting and output possibilities. The SQL queries can be performed interactively from both graphic and alphanumeric terminals. With the GRADIS extended SQL,

1 Nievergelt, Hinterberger, Sevcik: The Grid File: An Adaptable Symmetric Multikey File Structure. ACM Transactions on Database Systems. Vol. 9,1, March 1984, pp. 38-71.

users can perform ORACLE SQL operations using spatial relationships under control of the GRADIS environment (either graphic or alpha-numeric) and place the results of SQL object queries in a buffer for display or for further processing.

GRADIS has a comprehensive capability to perform extensive algebraic set operations on object data stored in either temporary or permanent buffers. The results of spatial queries and polygon overlays can be saved as permanent sets or as temporary sets for additional processing and other logical set operations.

Thematic analysis can be displayed on the screen and can be stroked to produce high quality plots. The display can be saved or discarded as desired.

GRADIS provides two methods of displaying thematic information:

1 - Logical object coverages or layers defined in the data model with their corresponding graphic representations is the default method of displaying general thematic information.

2 - The displaying of thematic areas based on attribute qualification can be performed with the GCL Enquire, Select, and Thematic Display commands. A set of graphic transforms corresponding to specific object attribute values can be defined in a table. Objects which match these attribute value when queried are assigned the appropriate graphic representation. This process results in the automatic creation of a thematic display.

2.4.2 Macro Command Language

Existing functionality can be user-customized and new functionality can be added as required using the GRADIS application interface, an integrated Macro Command Language facility. The Macro Command Language (MCL) has a "learn mode" that allows users to create new functions simply by recording a sequence of interactive operations within a GRADIS working session.

MCL is an extension of the GRADIS Command Language described above and is used to combine repetitive sequences of commands with mathematical and logical operators into one named command. Embedded SQL statements also can be included in macros. With MCL, the user can extend GRADIS with application-specific user interfaces to tailor and considerably simplify working procedures. Accessing external applications from GRADIS is accomplished by using the MCL shell command and the MCL buffers. The MCL shell command provides access to UNIX commands, shell scripts, and compiled programs. The MCL buffers provide temporary storage to send and receive data between GRADIS and external programs. For example, the user can select objects with the SELECT command and store the results in an MCL buffer. This buffer can be sent to an external program fro processing. Upon completion of the processing, the external program writes the results back to another MCL buffer. This processed information can then be displayed or highlighted with standard GRADIS GCL commands.

2.4.3 The GRADIS User Interface

The acronym GRADIS is derived from a long line of mapping products originally from Contraves AG and stands for Graphical Dialogue System. GRADIS uses the X-Windows Menu System and OSF/Motif guidelines for UNIX workstations as a standard interface.

The GRADIS system is command driven with sophisticated commands directly accessible to users in a manner similar to the command level of an operating system. Users with computer experience will appreciate this direct access to system functionality. It provides a very flexible way for system administrators and their technical staff to quickly access internal functionality for routine system management. Technical personnel can also build custom applications for general users by combining repetitive sequences of these commands with logical operators using the Macro Command Language. But most users require easier access to GIS functionality. To make this easier access possible, these commands also can be selected via menus with the GRADIS Graphical Users Interface (GUI). The GUI has been designed specifically for ease-of-use and customisation. With the GUI, users don't have to remember command names: they are presented directly on the menu and options appear automatically in sub-menus. The definition and creation of the GUI can be divided into the following two parts:

1 - **Menu definitions files**: A large part of the GUI is defined in a series of ASCII files called "menudefinition files". The GUI element definitions include the menu bars with their pulldown and sub-menus, and various types of button and toggle fields. These files are "compiled" by a processor for run-time. Named boxes can be added to the menus as desired to substitute system command and function names for those that users may more readily recognize.

2 - **Dialog boxes**: A series of "dialog boxes" are included with the GUI to simplify and guide users through

interactive operations. These are separate windows (popups) that are displayed or removed from the screen as necessary. They are especially useful for guiding users through detailed sequences of commands performing more complex functions.

The GRADIS GUI can be flexible altered to meet specific application requirements or operator levels by easily changing a wide range of X-11 resources such as colors and fonts, native language definitions, and the general layout of the menu interface. The GUI logically subdivides the system's command components and provides a logical flow and context for all operations.

CONCLUSION

GIS is a major investment in new technology for data capture, data storage, and data analysis. It is an enabling technology and a new and better way of doing things for a variety of organizations and disciplines whose activities and end-results depend on the effective analysis of geographical information. The introduction of GIS into organizations is not trivial because of a variety of human, organisational, and technical factors that must be considered for successfull implementation. But the benefits are substantial. Once introduced, a GIS provides organizations a comprhensive and cost-effective means of accessing, sharing, and analyzing geographic data for decision support. The longevity of data and the interdisciplinary use of data are the most important technical factors to consider when selecting a GIS because these factors are directly related to cost-benefit. The four architectural principles of GRADIS fulfill these important requirements while providing extensive GIS functionality for present and future user-specific applications.