CREATING USER FRIENDLY GEOGRAPHIC INFORMATION SYSTEMS THROUGH USER FRIENDLY SYSTEM SUPPORTS

Powel Crosley
Data Research and Applications, Inc.
412 Cedar Bluff Rd. Suite 203
Knoxville, Tennessee 37923

ABSTRACT

For over twenty years, geographic information systems have been growing in stature as more and more firms and governmental agencies realize the need for automated methods to retrieve, analyze, and display geographic data. This has been paralleled by a growth in the number of information systems available, and consultants to develop and/or manage them. Technical applications have evolved to the point that it is sometimes user friendliness which sells one system over another, rather than what can be produced by either system. This paper will explore the concept of user friendliness in geographic information systems from the standpoint that a truly friendly system has to have friendly 'supports' on which to build. These 'supports' include a database management system, a device independent graphics package, a human-machine interface, and a highly available computer system. Following a discussion of these supports, a brief description of a prototype system utilizing these 'supports'.

GEOGRAPHIC INFORMATION SYSTEMS

Many professions have developed automated methods for assisting in data analysis and decision support. Terms such as 'decision support system' (DSS), 'management information system' (MIS), and 'interactive information system' (IIS) are commonly discussed and closely related. So too are geographic information systems (GIS), which will be defined here as a set of procedures that support the acquisition, manipulation and use of spatial data. Until recently there was no interaction between these fields to try and utilize or develop concepts which are in fact common to all the above systems [Johnson, 81].

There are obvious differences in the nature, purpose, and requirements of different forms of information systems. In geographic information systems the primary difference is its use of spatial data. Spatial data can be defined as any attribute, entity, or identifier which can be referenced to a geographic place. It must be recognized, though, that geographic information systems are just another flavor of decision support systems, and face many problems and considerations common to its non-geographic counterparts.

USER FRIENDLINESS IN INFORMATION SYSTEMS

Regardless of the type of information system, all decision support systems have one common criteria; a well designed, "friendly" user interface. It is not uncommon to see over 50% of generated code devoted to the user interface in non-geographic systems. In contrast, geographic information systems often devote 30 - 35% of their code to a user interface [Nicholson, 83]. Does this mean that geographic information systems are inherently less user friendly than their non-geographic counterparts? No! What this indicates is that "user friendly" is one of the most ambiguous criteria faced by system designers.

What Is User Friendliness?

While concrete definition may be difficult to arrive at, there are several attributes of user friendly systems that can be generally accepted. First, system directions or prompts should be understandable by non-technically oriented personnel. Second, the system should be flexible enough to provide shortcuts for experienced or technical personnel. Third, error messages should be clear in their meaning, and provide some direction on how to correct the problem that occurred. Fourth, the system should be able to handle different approaches to the same query. Fifth, there should be verbose directions under a 'help' umbrella, to fully illustrate what is expected of the user. Sixth, the system should be available when the user wants to use it.

These attributes are not all inclusive, but illustrate some primary areas where systems commonly have problems. Designing and developing user friendly systems is an evolutionary process. Obviously today's interactive systems are much more friendly than the batch oriented systems of days gone by. Even today, though, the person accessing the system is often not the decision maker but rather a middleman, possibly because of the decision makers concern over his ability to extract what he wants from the system. This will probably change in the future because today's middlemen will become tomorrow's decision makers, but that in and of itself will not mean that systems are more friendly. The continued evolution of user friendly systems depends upon incorporation of new hardware technologies, and software methodologies.

DEVICE INDEPENDENT COMPUTER GRAPHICS

A common occurrence in earlier geographic information systems was the constraint of what graphics device could be utilized to view system output products. While this particular constraint helps the turnkey vendor who offers a total system, it negatively impacts a buyer who has similar existing equipment and limited budgets. Dependence on specific devices or protocols also leads to major software rewrites when display devices become obsolete, or new display technology calls for upgrading the existing system.

Since the early seventies, this problem has led to attempts to develop and standardize device independent computer graphics software [Scott, 84].

What is Device Independence?

Device independence is the characteristic of a graphics support package which allows application programs to run on different types of display devices [Warner, 81]. For instance, a program which generates a map utilizing solid fill colors on a Tektronix (or any other) raster display device can create the same image on a Hewlett-Packard (or any other) pen plotter without any change to the source code. This is accompilished by the graphics support package, which maps the application program graphics requirements into the protocol of the intended display device. This protocol mapping is handled by individual device drivers.

Device independence decreases the impact that device obsolescence has on geographic information system development. It provides the developer with a set of graphics tools which allows the system to address a virtual, universal display device [Scott, 84]. Buyers of device independent GIS technology also benefit from the abilty to incorporate any hardware they may already have, as an integral part of their system.

Standards in Device Independence

Standardization of device independent graphics support packages is an obvious necessity to insure transportability of developed software, and programmer skills. Currently, several standards either exist, have been proposed, or are in development. These standards include the SIGGRAPH Core, Graphics Kernel System (GKS), North American Presentation Level Protocol Syntax (NAPLPS), and Programmer Hierarchical Interface to Graphics Standards (PHIGS), as well as the Initial Graphics Exchange Specification (IGES) [McLeod, 84; Scott, 84; Warren, 84]. Soon it should be possible for entire geographic base files and graphic databases to be transferred from one vendor system to another. Imagine the time and cost savings that could be realized if your entire graphic database which was developed on system 'X' can be directly installed on system 'Z'.

FAULT TOLERANT PROCESSING

Throughout the vast array of automated information systems, some can be seen as 'heartbeat' applications. In other words some systems have to be constantly running, highly available, or some serious consequences will result. Obviously these types of operations can not afford the slightest amount of down time. In less critical systems down time can also pose serious consequences if the system is not available when you need it. How many times has a demonstration been scheduled only to be scrubbed or postponed because the 'computer is down'? How much programming time has been lost due to down time on development systems? The need for fault tolerance then not only applies to 'heartbeat' systems, where continuous processing is critical, but also to any system where availability is important at any point in time.

What is Fault Tolerance?

A fault tolerant computer system is one in which any single failure is transparent to the user. To be considered fault tolerant the following characteristics must be present. First, each critical component must be replicated to allow replacement after failure. Second, the system must be able to identify a failed component automatically. Third, the failed component must be isolated electrically and logically from the rest of the system. Fourth, the system must reconfigure itself to continue processing uninterrupted. Fifth, the faulty component must be repairable without disrupting ongoing system operation. Sixth, once repaired the component must be reintroduced to the system. Finally, no component failure should allow the database to be corrupted [Highleyman, 84].

Until recently there was only one fault tolerant computer system on the market. In the past five years, though, over a dozen new systems have become available utilizing different means to the same end [Serlin, 84]. To illustrate how fault tolerant systems operate, consider yourself working away on a machine that appears to be operating normally. Perhaps what you don't see is that one of the duplexed CPU boards malfunctioned, which triggered an automated call from your system to the central support center across the country. There the trouble was logged and a spare part placed in overnight mail to your office. When the part arrives the operator simply opens the cabinet pulls out the failed board and slides in the replacement, while your important demonstration continues uninterrupted.

DATABASE MANAGEMENT SYSTEMS

The heart and soul of any geographic information system is the database management system (DBMS). In general, all database management systems serve to separate the user's logical view of the data from the physical organization of the data. The differences between hierarchical, network, and relational systems are mainly the way they allow the user to view the data, and the way they map that view into the physical organization [Martin, 77]. Within the past few years relational database management systems have become more widely accepted as vendors and buyers see some inherent benefits in the relational model over the network or hierarchical model.

What is a Relational Database Management System?

There are several criteria for labeling a DBMS as relational. First, the database is represented in the form of two-dimensional tables. Second, there are no navigational requirements imposed on the user. Third, any number of tables may be projected or joined [Venkatakrishnan, 84]. In terms of applications, the benefits of a relational DBMS can be summarized as follows:

application programs are smaller, and easier to develop; fewer application errors; shortened development time; ability to handle ad hoc queries; and ease of expanding data elements and tables. The relational DBMS is ideally suited for applications where queries are not well known ahead of time, because retrieval schemes and logic do not need to be understood by the user [James, 84; Wood, 84].

Neither traditional nor relational data models are going to be able to efficiently handle all the requirements of a geographic information system. However the wide range of queries that might be asked of a GIS, and the need to be able to handle new layers of information, or modify existing ones, clearly indicate a relational model should be the basis of geographic information systems.

HUMAN - MACHINE INTERFACES

The most visible aspect of any information system is its user interface, how the machine communicates with the human user. Not surprisingly, it is also the most common area where users have problems with a system. Users generally have a good idea of what they want from the system, and a general understanding of how to instruct the system to perform the task. Everyone, though, occasionally commits errors, needs extra information, or has a lapse in memory. When these situations occur the system should be a forgiving friend to the user, helping them back on track. What too often happens is the messages or actions taken by the system are unclear in meaning, and further confuse the user raising their frustration level.

Forms Management Systems in Geographic Information Systems

A contributing factor to user confusion can stem from the fact that different application modules were developed by different programmers with different levels of sympathy for the user. One module may have well designed, explicitly worded prompts and error messages, another may have a tersely worded prompt or error message with ambiguous meaning [Norman, 83]. The incorporation of a forms management sytem (FMS) can limit some of these problems because it provides uniformity in what the user sees on the screen, usually in the form of menus. A robust FMS will include capabilities for explicit help screens, error messages, look back features, and terminal independence. In addition to assisting the novice user, FMS packages typically allow random traversing of the menus. This is a benefit for the saavy users who know exactly what they want and what menus they need to answer [Mason, 83].

Another feature of robust FMS packages was briefly noted above, terminal independence. Similar to device independent graphics software, terminal independence allows a user to work with the system utilizing any terminal type he may have available. The application program is developed to utilize a virtual terminal, and the FMS translates the virtual commands into the volcabulary of the target device. This feature is terrific for information sytems with time sharing users in remote locations. If the user has a terminal that uses TTY protocol, they can access the system just as well as an onsite user who may have VT100 protocol.

BUILDING A USER FRIENDLY INFORMATION SYSTEM

Utilizing the previously discussed 'supports'; device independent graphics; relational database management system; forms management system, and fault tolerant processing; it is possible to develop a more totally friendly geographic information system. A system which can minimize initial costs for display peripherals; a system that can handle ad hoc queries with no a priori knowledge by the user; and a system that is available when it is needed. To illustrate, the following description details the purpose, requirements, and solutions for a geographic information system with parcel level resolution both for the geographic base file and attribute information, for which the prototype currently exists.

Development of the GIS was commissioned to provide one immediate capability, and capacity for some specific enhancements to enable broadening the user base at a later time. The immediate capability dealt with allowing a distributed user network to query a geographic data base for prior title policies on specific parcels of land. Expanded capacity was specified as including capability for demographic analysis, tax appraisal, title exceptions and property liens, and structural attributes for realty evaluation.

Immediately, it is apparent that the database will require extension and modification in the future. Queries will be ad hoc in nature and may need to aggregate parcel data to block or census tract level, or retrieve census data based on title policy numbers. After reviewing several DBMS packages based on network and relational models, it was clear that the relational model was the one that could most efficiently manage the data to be resident in the system.

In terms of peripherals, most users or potential users would already have some type of terminal or personal computer. The need to minimize additional hardware cost was an explicit requirement of the finished system. Even though most of those existing terminals are non-graphic, a large majority of them can be retrofit with a graphics board to emulate one of several graphics devices. Device independent graphics software is the only sensible alternative in this case, because it reduces system development time, and allows the user to maintain his familiar equipment.

Menu driven prompts was another requirement for the user interface. Most existing users had experience on a time sharing non-geographic system which was to interface the finished GIS for providing title information. This existing system was menu driven, but not by a FMS. A design decision was made to utilize a FMS package to aestheticaly emulate the structure of the existing menu system, provide terminal independence, and reduce development costs.

Finally, the system is expected to be heavily utilized and system availability is a must. The prototype system has therefore been developed on a fault tolerant processor to eliminate system down time. While it appears at first glance that fault tolerance would be prohibitively expensive, in fact the difference between a DEC VAX processor and some equally powerful fault tolerant systems entails only about a ten percent premium for fault tolerance. How often would you have been willing to pay ten percent more for a system that would not go down at all the wrong times?

CONCLUSIONS

User friendly geographic information systems will never be 100% user friendly, because everyone has a different definition of user friendly. It is currently possible, though, to develop systems that:

- Do not require a priori knowledge of database navigation paths,
- Do not require purchasing of peripherals for which a user already has similar equipment in place,
- Do not require cryptographic translation of error messages or responses to 'help',
- Do not require users to use the excuse, or operators to invent explanations why, 'the computer is down'.

A geographic information system with these features is surely a positive development in the continued evolution of user friendly systems.

REFERENCES

Highleyman, W. H., 1984, "The coming day of fault tolerance":Computerworld, September 10, pp. ID/1 - ID/11.

James, B., 1984, "Network vs. relational: Fit the solution to the need": Computerworld, September 24, page SR/38.

Johnson, T. R., 1981, "Evaluation and Improvement of the Geographic Information System Design Model": an unpublished Masters Project, Department of Geography, State University of New York at Buffalo.

Martin, J., 1977, Computer Data-Base Organization: Prentice-Hall, Inc., Englwood Cliffs, New Jersey.

Mason, R.E.A. and Carey, T.T, 1983, "Prototyping Interactive Information Systems": Communications of the ACM, Volume 26, No. 5, pp. 347 - 354.

McLeod, J., 1984, "Graphics Capabilites Abound": Systems & Software , Volume 3, No. 8, pp. 82 - 94.

Nicholson, R. L., 1983, "Mapping for Decision Makers": Computerworld , November 28, pp. ID/43 - ID/48.

Norman, D. A., 1983, "Design Rules Based on Analyses of Human Error": Communications of the ACM , Volume 26, No. 4, pp. 254 - 258.

Scott, G. M., 1984, "Practical Implementation of GKS": Computer Graphics World, Volume 7, No. 12, pp. 35 - 38.

Serlin, O., 1984, "Fault-Tolerant Systems in Commercial Applications":Computer, Volume 17, No. 8, pp. 19 - 30.

Venkatakrishnan, V., 1984, "Handling data integrity in relational DBMS": Computerworld , October 8, pp. ID/25 - ID/30.

Warner, J. R., 1981, "Principles of Device-Independent Computer Graphics Software": IEEE Computer Graphics and Applications , Reprint from October Issue, pp. 1 - 10.

Warren, C., 1984, "Graphics Software Schemes Enhance Peripheral Interfacing": Mini-Micro Systems , Volume 17, No. 9, pp. 163 - 178.

Wood, D., 1984, "Emphasis upon physical data structures can distract from accurate analysis of relational DBMS makeup": Computerworld , September 24, pp. SR/12 - SR/13.