Integrating Mechanisms for Storage and Retrieval of Land Data

by Andrew Frank

Introduction

Traditional manual land information systems (LIS) are notorious for problems with information dissemination. It is necessary but difficult to make copies of maps and registers and keep the collected data up-to-date. This is especially problematic if multiple copies must be updated manually. Computers may help overcome these limitations, as they allow one to copy, massage, and disseminate information easily. With the help of electronic data processing, it is possible to integrate data collected and stored in different registers and to avoid many of the limitations inherent in slow and expensive manual processes.

Although a number of these systems have been designed, many of the more polyvalent ones failed to achieve their goals (Mead, 1981); systems built for narrowly defined applications have sometimes worked well until the requirements of the application changed. It has proven extremely difficult to make the integration of data for different purposes work.

Integrating data from different sources and making them available to different user groups, a central concept in the LIS philosophy, is considered here. This paper discusses why integration of data for different purposes is difficult and what research must be done to enable us to build multipurpose cadastres as advocated in the NRC (National Research Council of the National Academy of Sciences) report (NRC, 1980).

This paper sets out to describe research topics which it is believed are important to achieving the goal of an integrated LIS in this and the next decade. It is as difficult to describe clearly prospective areas of research as it is to chart a map of "terra incognita"; the unknown cannot be clearly defined. Thus, this paper can mark the problem areas but must be sketchy in outlining the means that can lead to solutions. The fundamental problems that plague present systems have been concentrated on and a time horizon of about 10 years into the future has been selected. To assess the important topics for research beyond the next decade seems very speculative; it is assumed that a rigid formal base as advocated throughout this paper will provide a worthy base for future endeavors.

Challenge

Building the systems that are perceived necessary and economically feasible in a timely manner is important for our society. Systems which provide information about land are increasingly important for decisionmaking in land use and planning and crucial for the imminent renewal of infrastructure installation in the United States. It is being asked that theoretical insight and practical knowledge be provided to design and build cost-effective multipurpose systems and avoid the "everybody his own prototype" approach, which is extremely costly and often disappointing.

Tools Available

A number of recent developments have provided new tools to attack the problems:

Recent Hardware Development

- Considerably faster processors have become available and today's personal workstations have the power of yesterday's mainframes.

Professor Andrew Frank is with the Department of Civil Engineering, University of Maine at Orono, Orono, Maine 04469

Surveying and Mapping, Vol. 46, No. 2, pp. 107–121

- Large storage spaces on hard disk are available at reasonable cost.
- High bandwidth communication media for short- and long-distance communication are available.

There is no justification to design systems which do not use the available hardware and excuse shortcomings with savings in the hardware needed. Cost for data collection, user time, and software is so much greater than for hardware.

New Software

- Multiuser, multitask operation systems are available even on smaller personal computers.
- New, improved programing languages and programing support systems have become reality (Ada, Modula-2, Pascal).
- Software for passing data from one computer to another and integrating several computers into a uniform system (computer network) is now available.
- Database management systems (DBMS) to provide integrated data storage space for data from different sources have been developed.
- Expert system support and nonprocedural languages for fast prototype and writing of small programs (fifth generation computing) are available.

Theoretical Advances

The following theoretical advances have been made in computer science:

- Formal methods to describe the function of programs and to apply strict mathematical operations to them;
- Mathematical theories for databases and related operations; and
- Approaches to a consolidated theory for distributed processes.

In the area of human factors studies, better understanding has been reached of ergonomics and human factors related to user interfaces for programs (user friendly programs).

In the area of information science, studies have been made into the nature of information and the representation of information in computers.

Conclusion

It is a challenge to the surveying engineering profession to learn to use these new tools to improve the means with which they provide land information to society. These tools make it possible to design and implement integrated LIS. It should not be attempted to build LIS without their use. However, it must also be noted that most of these new tools have not yet been applied to problems of the complexity of an LIS and their application will need considerable work.

If surveying engineers do not pick up the challenge and improve their services, they will not remain competitive in the future information business (Naisbitt, 1984) but will be reduced to being the measuring "gophers" for others that know how to handle and market the information systems desired.

Organization of the Paper

The framework of this paper is "integration" in a land information system. We have to deal with integration of different user needs, data from different sources, data stored in different computer systems, and programs for different tasks written by different programers.

This paper concentrates on seven domains, where research efforts are believed to be most necessary, namely:

- Layered framework for a computerized LIS and identification of major modules;
- Improved data description methods, including data encoding and data quality;
- Treatment of geometric descriptions, including problems of limited precisions;
- Integration of temporal data;
- Requirements of data retrieval and query language;
- Organizational concern of importance for the design of databases; and
- Technical solutions for data storage and retrieval.

For each we have given some explications and specific recommendations for studies of various size.

Assumptions

This paper assumes that in the next decade there will be no major changes in the economic, social, political, or technical environment.

It also assumes that development of computer hardware and computer science will continue along the lines presently observed.

It is assumed the LIS to be built will be integrated, providing data from different sources to different users. It is also assumed they will be interactive, providing immediate response to the user. The advent of the personal computer has shown users how fast and convenient interactive systems are, and users will not easily settle for the batch system which delivers its result to a central location "tomorrow."

Integration of Different Needs as the Principal Problem

Size and Complexity of Necessary Software

The size and complexity of an integrated LIS is a topic worthy of research. The effort to construct a program increases faster than the size of the program. To write a program of twice the size requires far more than twice the amount of time. A multipurpose LIS typically needs a much more complex program than a single purpose information system and cannot be built by combining software individually built for single purpose systems.

In writing large integrated software packages, special tools for design and analysis are needed. If these tools are not available, chances for successful completion within the planned budget become slim. For instance, who would contract a number of carpenters to build a large hospital complex without any further planning and guidance?

Necessary Theoretical Understanding of Application

Programs to solve problems that are not well defined and understood cannot be written. Unfortunately, it is often tried. On the other hand, software currently being developed is forcing us to ask important questions about a task and record the findings in a formal way. Thus, constructing prototype software together with formal methods of software engineering is an effective tool for investigating certain tasks and exploring the correctness of the results. Nevertheless, explorative writing of prototype software should always be separated from later design of production programs.

Layered Design

Software engineering proposed to split large systems in layers with specific functions. Each layer would provide some specific services which are then assembled and improved by the next higher layer to provide, in turn, higher level functions. The best known application is in the "open systems interconnect" model for communication between computers proposed by the international organization for standardization. Examples for the application of this idea can be found in Frank (1981, 1984) and Chevallier (1984).

Modular Software and Identification of Common Functions

Software complexity can be reduced by building large systems from modular building blocks. If a module fulfills a specific function with minimal interactions with other modules, overall complexity is reduced. Modules are written independently of others, and only the defined interface of the module is of concern to the programer who writes a piece of software using this module. The technique is essentially a method of abstraction because the details of a function are hidden inside the module and the writer of the using program does not need to know them. An additional benefit is that changes in one part of a program do not propagate in other parts of the system.

Software engineering advocates, under the name of object-oriented programing, a method to group together the data describing an object and all the operations related to these data. This concept is independent from the programing language used. The new languages, like Ada and Modula-2 (Wirth, 1984), incorporated related concepts, but the traditional languages, like Fortran (Isner, 1982; Aronson, 1985) and Pascal, also can be used.

The method is related to the methods of specifications based on abstract data types (Zilles, 1984; Gougen, 1978; Parnas, 1977). The data objects defined are likened to the data types and operations the computer hardware offers. Good software engineering practice demands that formal descriptions of a system be written before the actual programing (coding) starts. This permits problem identification and correction of errors in the earlier stages of the design and avoids costly changes when much of the code is already committed.

Formal specifications of this type can be considered as defining an algebra (multisorted or heterogeneous algebra), making them subject to mathematical proof methods. This permits carryover of ideas from algebraic theory (like the algebra of real numbers). The algebraic treatment helps to group, identify, and relate functional components when analyzing user requirements and current manual procedures, and helps to provide a formal framework for strict reasoning. Currently several groups are working on automated systems to help designers use those techniques. Such tools should be used as soon as available.

Software Packages To Support LIS in General

Software for a multipurpose system is not the sum of the programs written for each individual purpose. Common routines must be identified; all routines providing a certain function are concentrated in a module and then this module is used everywhere these functions are needed. This is the basic idea behind database management systems, database query languages, and graphics packages, but the same idea can be applied further to packages typical for LIS (but not found in other information systems) which can be written independent of a specific purpose of an LIS. Such packages can be written in a transformable manner, so they are available to users at different sites. Generally it seems desirable to make such packages small (i.e., including as small a group of functions as possible) and independent of each other to facilitate changes and adaptation. It may be worthwhile to set up an organization for exchange of software packages among research groups and standardize the software environment (operating systems, languages, etc.).

To design such packages, increased understanding of the functions in a multipurpose LIS is needed. To this end, more user requirement studies are not advocated, as those will reveal only the superficial traits of users' needs, but in-depth and formal studies of the basic functions as outlined in the sequel to this paper are. Our present understanding of the interaction between user and data processing systems, and the translation from data to information is not far enough advanced; the absence of a theory allows no more than descriptions of how users imagine their present systems work. Often user accounts do not include exceptional procedures and error treatment, two critical ingredients in any automated system, which are hard to add to the design after it is completed. Techniques from artificial intelligence and communication theory can be applied to describe actual performance.

Formal methods for description of operations provided in a package and related fast prototyping methods with which to experiment will help considerably to increase understanding of the processes involved.

Recommendations

It is recommended that a study be undertaken to describe a layered operational framework for the different software modules which can be identified in an LIS. This study should use modern software engineering methods based on formal mathematical methods (multisorted algebras or other forms or formal specifications languages). It must describe the contribution of each layer, its operation, and how it used the layers underneath.

The study should identify conceptual models which are generally useful for the construction of LIS software and should describe their function and their interfaces with each other.

Formally specified and documented software packages should be written in a transportable form and made available to research groups.

Data Description Method

Integration of Data from Different Sources

The data for an LIS likely will not come from one single source and most of it may not be collected with multiple uses in mind. The strength of an integrated LIS is its ability to exploit data from different sources in a joint mode (which does not necessarily imply that the data are stored in one single location). However, it is necessary to have a logically integrated description of the data, which will consist of a number of levels.

Entity, Attributes, and Relations

Conventional databases are based on a data model that uses entities, attributes, and relations (using varying terminology (Chen, 1976; Codd, 1982)) for data description. An entity, comparable to a record in conventional data processing but also to an object in an object-oriented setting, has attributes that further define the entity and may be related to other entities of the same or different type. Using these categories, it is possible to describe a data set in one or several files or in a database. The method is formal and capable of precise description that rules out certain misinterpretations.

Generalization

The traditional methods for data description are insufficient for LIS because an LIS very often contains data about objects that are further subdivided into more special classes; e.g., buildings are divided into residential, commercial, industrial, farming, etc. Treatment sometimes deals with the general concepts of building; sometimes only some of the subclasses are relevant. The concept of generalization must be added to the data description method. It is the same idea that in artificial intelligence (AI) is often called an "is a" relation (or "specialization," if considered in the other direction) (Borgida, 1984). With this extension, schema description methods become closely linked to semantic nets (or frames (Finder, 1979)) in AI systems; the relevant literature should be more closely followed by the LIS community to detect results that become applicable.

Classification and Encoding

The state of reality must be perceived and represented with symbols having an agreed-on semantic; e.g., taking the measurements that point A is 7.35 m from point B and writing it into a deed or writing that the house at Grove Street 30 is a two-story building.

It is not critical what symbols are used to encode the information, as long as there are well-defined methods to translate without loss of information from one system to the other (e.g., distance expressed in the metric system into the traditional American system). This is especially true for electronic data processing, where internal forms may be quite different from the values to which users are accustomed (binary, hexadecimal, etc.)

A difficult and little understood issue, however, is the selection of scale used for encoding (e.g., the assessor's scale might be one-, two-, three-story houses, or one-, one-and-a half-, two-, two-and-a-half-, etc.).

Encoding of attribute values can be classified into the following four groups:

- Nominal (e.g., residential, industrial, business zone);
- Ordinal (e.g., very steep, mild slope);
- Interval (e.g., in SAT scores, a difference in 50 points does not have the same significance at 600 and at 800 points);
- Ratio (measurement with base variable and scale).

It has been noted (Robinson, 1985) that users typically need the ordinal values, whereas operations are usually simpler on the ratio scale. It seems generally advisable to collect and store values on interval or ratio scales and then translate them into nominal or ordinal values for presentation to the user. Such a translation may use more than one attribute and combine them into a single information. For example, a user might be interested in the usefulness of lots for building a supermarket. The system would combine a number of attributes like accessibility number and socioeconomic status of the population within a 15-minute driving distance of the lots.

The data description in a database schema does not include description of the encoding system. There is a need for a method to describe encoding methods formally and to translate between them. Expert system methods may help to build translation routines easily and rapidly.

Data Quality

Not only can the encoding differ among data sources but so can data quality. Included in data quality are exactness (precision) as described by the standard deviation associated as a result of a measurement process, coarseness of encoding (encoding error), timeliness of the data, definition of reference object (see later geometric problems), and reliability (how many gross errors). Different aspects of data quality can affect different user groups.

Data Integration from Different Sources

The original description of data collected for different purposes may use similar terms to describe the observed objects, but their meaning based on different intentions may vary. For example, road width to a traffic engineer means the width of road used for traffic; in a hydraulic study road width is the width of the hardcover surface, and to the surveyor it is the width of the right of way.

It is crucial in building a multipurpose LIS that similar semantics be detected, in spite of the use of the same terms for different names and dissimilar semantics.

The techniques for data decriptions can be used to describe data from a single source, data for a specific purpose, and all data available in an (possibly dispersed) LIS. To combine data from different sources, the data from each source is described and then the different descriptions compared to identify commonality. Artificial intelligence has studied methods to detect similar semantics in concepts in spite of dissimilar terms used (Finin, 1984). In the database literature, some procedures for automatic detection of overlap in the data descriptions of different user groups (Kent, 1981) are found.

Standards for Data Exchange

The standardization of formats for data exchange is not an imminent problem. Loading data received in a sequential format into a database is much more affected by differences in the classification and encoding methods and the data quality than by differences in the arrangement of fields within the records. This is significantly different from exchange of cartographic base files, where programs operate directly on the files provided and the structure of these files is crucial for their operation. It may be attempted to define a simple language to describe the contents of a file and design programs that automatically load data from such files in a suitably structured database.

However, we expect the studies listed above to produce tools that allow descriptions of the semantics and the quality of the data after translation and help to overcome differences on this level.

Consistency Constraints

Not all combinations of data form valid models of reality (e.g., persons less than ten years of age cannot have children). In large data collections, it is of geat importance to describe and enforce such constraints. This is advantageous for detecting errors in the data as early as possible and possibly more important for guaranteeing certain invariants for processes operating on the data: An algorithm has to assume certain properties of its input data and cannot produce reasonable results if these are violated. It is possible to write an additional code which tests the properties explicitly before executing the algorithms, but this is generally a wasteful solution. It is more efficient to describe those consistency constraints formally and guarantee that the data stored comply. This needs to be checked only during operations that change the stored data.

Recommendations

It is recommended that a study be undertaken to find, assemble, and publish formal methods for a most complete description of data as they are likely to be found in an LIS. This study should be based on the abstraction mechanism identified in recent artificial intelligence studies and explore how data description methods can be expanded and integrated to include more of the meaning associated with the data from different sources. The methods must take into account and be able to describe consistency constraints of all types. They should be tested in an example LIS and then widely disseminated to encourage practical use.

It is necessary to analyze the different classification and encoding methods and their relative merits. Work has to go beyond the encoding theory and deal with the semantics involved. It should result in recommendations for the selection of encoding methods and investigate conversion from one code to another.

Further study should be undertaken to identify and classify the different aspects of data quality, provide methods to describe them, and explore to what extent they affect the use of the collected data for certain user categories. Research should also explore how to combine data with different qualities to obtain useful results.

Integration of Geometric Data

An LIS deals primarily with data related to land. Each data element is related to a reference object it describes. It is thus necessary to discuss methods to describe the geometric properties of reference objects and also to find methods to combine descriptions of different objects exploiting their known geometric reference. Those methods may build on and use the methods for data description given above since some of the problems encountered are similar (e.g., data quality). Geometric data, however, is of such overall importance that special and more detailed treatment seems warranted. A mathematical "Spatial Theory" has been listed as one of the most crucial research needs for geographic information systems (Boyle, 1983).

Reference objects which are either geometrically exactly defined (independent of the precision with which the limits are known) or are only generally defined without any clearly determined boundaries. Parcel boundaries are of the first class (i.e., always an exact boundary, even if not presently known), whereas natural limits often fall in the second category (e.g., the exact limit of a forest, of South Carolina's "low lands," or of the permafrost areas are unknown).

Geometric Description of Exactly Defined Objects

The exact geometric definition of an object does not automatically include precise knowledge of this geometry. In fact, all measuring methods can only be approximations of "true" values, due to inherent measuring errors.

A number of specialized data structures to treat and store geometric descriptions of specific objects have been developed in the past. Best known is probably the DIME encoding method used by the Bureau of the Census for encoding of census tracts. Independent descriptions of object of geometry by closed polygons are also commonly used.

However, these methods are useful only for the specific geometric structure of the selected objects treated. In "Conceptual Framework for Land Information Systems—A First Approach" (Frank, 1982), geometric structures were classified from a few topological criteria used as a base for classification of the geometric structure of specific object classes and their geometric consistency constraints. Similar considerations can lead to algorithms to treat and store geometric descriptions.

Recent work (White, 1984) has brought more attention to the problems caused by finite approximation to exact geometry due to measuring errors and limited precision of computer arithmetic. It has been shown (Franklin, 1984) that there are no workable alternatives to use of approximations for geometric computations (computer reals), and it is concluded that the integration of topology and metric positions is difficult; they may be treated in parallel, but separately.

Attention should be directed not only to an efficient storage of geometric descriptions but also to the corresponding geometric operations. Geometric data must be manipulated to change them, deduce other data of interest, or represent the data. Computational geometry has partially explored this field and numerous algorithms have been published. "A valid criticism is that many of the algorithms appear to be difficult to implement in practice or have unrealistic restrictions. A bugbear to the implementer is the plethora of special cases which tend to be glossed over in theoretical algorithms" (Forrest, 1984). The design of a generally useful datastructure must take these problems into account and support the most important basic geometric operations necessary for an LIS. The LIS must maintain the geometric consistency constraints efficiently because algorithms from computational geometry are typically very difficult to "prove" against errors in the input data.

Representation of Objects with Imprecise Geometry

It might be useful to study the difference between objects for which geometric descriptions can only be given approximately and others for which descriptions could, in principle, be given exactly but are not known at the present time.

The definition of inexact boundaries is most often based on the combination of a number of factors (e.g., several components of climate exposition or altitude). It may be possible to use a deduction rule in an expert system to find most appropriate limits. Those limits, however, must still be considered as fuzzy, should be presented to the user as such, and further deduction must take this uncertainty into account.

Aggregation and Generalization

Generalization is used in cartography to reduce the details of an object to be rendered on a map of a smaller scale. Aggregation is the process in spatial statistics when values for a number of spatial units are lumped together to form a value for a larger spatial entity (e.g., aggregate all census block values to a value for a town). This can be understood as two occurrences of spatial application of the very general abstraction mechanism "generalizations." Obviously, spatial reasoning uses this method frequently; political subdivisions are an example for a hierarchy of generalizations; town, county, state, country .... Similarly, we can observe a hierarchy for linear features (streets, rivers) and for morphological features. It must be assumed that spatial generalization is an important part of spatial reasoning and must be studied and formally defined. Some initial work was reported in "Symbolic Feature Analysis and Expert Systems" (Palmer, 1984) and should be continued.

Recommendations

It should be studied how geometric decriptions of different objects can be treated and stored. The solution must be useful for objects of widely varying types and cope with how the limited precision positions are determined as well as the resulting differences in coordinate values for identical points (and in consequence, also for lines).

It should be studied how such a method could be based on measurements in lieu of calculated coordinates. Observed measurements are statistically estimatable and thus may allow for a statistical solution for the problems introduced by inevitable measuring errors.

The study may assess the relative advantages of designing specialized structures to treat large classes of geometric descriptions (e.g., partitions, communication networks, or data for digital terrain models) compared with a more general solution.

Further it is recommended that a study be made about the treatment of inexact (or fuzzy) geometric data. It could start with observations of how human experts (e.g., planner, geographer) use such data and what rules they apply. Those rules must then be formalized and generalized. This can be expected to lead to a better understanding of semantics and, then, to reasonable operations that can be programed.

Treatment of Temporal Data

Large parts of reality are that LIS models change in time; values describing this reality are thus valid only for a specific moment. In traditional manual systems, those problems were often explicitly or implicitly dealt with by annotating values with the date of their collection (e.g., by explicitly dating them or by implicit reference to some other dates). Human users automatically take into account possible changes in the timespan since the data collection.

Many user groups are not only interested in the present values (what is there now) but in the changes in the past that led to this situation. This is of prime interest to the American legal system for all operations affecting legal rights, but it is also of importance for planning and scientific purposes.

Only recently the database community has begun to investigate methods to deal with temporal data and apply the results of investigations in temporal logic by mathematicians.

Annotations of Data Values with Their Collection Date

Different proposals as to how time information could be connected to data values can be found in the literature about nonstandard database applications. They were made primarily for CAD/CAM (computer-aided design/computer-aided manufcturing) applications which have slightly different requirements.

Recording of Dates of Changes in the Data

Changes are not always reflected only by changing values for certain attributes of an entity (e.g., the name of the owner of a parcel) but may cause the end of certain entities and a contemporaneous creation of one or more new entities (e.g., during the subdivision of a parcel).

It seems insufficient to record those

facts separately, only dating each of them. An investigation of whether or not the changing operation itself should be recorded is recommended. This is almost analogous to the recording deeds in the legal system. It also seems to have theoretical advantages because an information system based on recorded changes is logically monotonic, whereas systems which record the presence of facts are nonmonotonic. It must be expected that the concept of recording changes may interact with the system selected to structure geometric data. Careful selection and design of basic operations on geometry may facilitate the dealing with temporal data.

Operations for Dealing with Time

Every method to record time must define operations to use this temporal information and define what is meant by these operations and their results. For example, what does it mean exactly for event A to be prior to event B? Temporal logic is a special sort of model logic (Rescher, 1971); it is important to base the treatment of temporal knowledge on a firm mathematical foundation.

Users will need to change data recorded as reality changes and to correct errors committed previously. It is especially difficult to decide what sort of changes to the events in the past are legal: none at all?; only error corrections? The data model in an LIS most of the time is behind reality and changes in reality are posted with a certain delay. This must be included in the treatment.

Database Support for Temporal Data

Records of time when data were collected or changes effective will be important in advanced LIS. It is thus paramount that the database management system supports those operations effectively.

Integration of User Requirements for Data Retrieval

If data from different sources are integrated into one LIS, it must be expected that different user groups need access to these data in different ways. Traditionally, for every use of the collected data, specific access programs were written. This is not desirable because it would require a multitude of programs which presumably would contain a large number of relatively similar routines.

A user group wishing to access the data in the LIS needs three functions:

- User query language to express its data needs and describe the desired form of presentation;
- A facility to extract the necessary data; and
- Facilities to present the data in a form understandable to the user.

This section will discuss these needs for data extraction (retrieval in the strict sense); later sections will treat the other topics.

Retrieval of data from databases is a two-step procedure: The user specifies abstractly what data are required and the database then delivers the data. Databases generally provide a relatively simple language for users to describe the data requested. Unfortunately, the currently available query languages are not powerful enough for LIS applications. Languages that are relationally complete (Codd, 1982) do not feature recursion and lack extensibility to include spatial operations on geometric data.

Retrieval Based on a Key or Attribute Values

Retrieval of data must be possible using a unique key (e.g., the street address of a building). It also must be possible to retrieve all data fulfilling some logically expressed requirements. Standard database methods provide such retrieval and they must be integrated into a DBMS for LIS application.

Access Based on the Spatial Location of the Referenced Object

In an LIS it is often necessary to retrieve all data for objects within a certain area. Such a retrieval must be possible and has to be executed rapidly. It is expected that the data structure selected for storage supports this special type of access.

Retrieval with Nonunique Keys

In LIS typically a number of "nearly" unique key attributes can be found. For example, street names are "nearly" unique keys within a town, since the same street name is seldom used twice. It seems reasonable to allow use of those "near" keys and have the system ask the user a clarifying question in those rare cases in which the key is not unique.

Use of Temporal Data in a Retrieval

It must be possible to ask, "As of what date or time?" Other temporal query facilities should be investigated for usefulness.

Use of Geometric or Topological Relations for Data Selection

In an LIS it must be possible to select data for retrieval based not only on attribute values (e.g., frontage larger than 30 ft.) but also on geometrical (e.g., all post offices within 3 mi.) and topological relations (e.g., all parcels adjoining parcel x).

It is presently not obvious what number of such relations is sufficient. An appropriate course of action would be to construct languages so that the user can build in a new relation when he needs it (extendible languages). A spatial theory will help to construct a small but powerful set of relations.

Retrieval Based on Transitive Closure

Geometrical and spatial properties are often used for retrieving all objects which are linked to a given start object, or linked to an object which is already linked. For example, given a forest parcel, find all its neighbors which are also "forest" and all their neighbors; this results in finding all parcels of a cohesive forest. This is often called transitive closure and is easily expressed using a recursive description: connected (x,y) if neighbor (x,y) or connected (x,z) and neighbor (z,y).

Translation of Terms Used for Data Description

Different user groups will apply different terms for the description of data they request. Such terms willl not only require simple translations (i.e., replace house by residential building) but more complex regrouping of data from the stored form into the structure requested by the user. Here again an extendible language with sufficient power (comparable to the power of an ordinary programing language) provides this capability for all future needs.

User Languages for Description of Requested Data

The background and interest of users in an LIS will vary greatly, as will their involvement with the LIS. It will, therefore, be necessary to construct different interfaces for different user groups. "Parametric users" who frequently need the same type of data will be accommodated best with forms to fill in the blanks. Users that seldom contact the system may desire a guided dialogue and extensive help facilities. Finally, sophisticated users with greatly varying needs will demand a very flexible query language and be willing to spend the time to learn its use.

The basic query language described so far can be used as a common ground for building these user specific languages. The different languages are translated into a common form and then executed. A package to help build new user languages to translate the requests would be generally helpful. This package should take into account the necessary specifications for the output form.

Integration of Expert Systems Capabilities

The functions desired in the basic query language are very extensive and the methods necessary to implement such a powerful query language are very similar to methods used in expert systems (also called knowledge-based systems) (Kerschberg, 1984).

In an expert or knowledge-based system, factual knowledge about the world is combined with rules human experts in the field typically apply to achieve more "intelligent" use of data. An expert system provides a framework to store facts and rules from different sources. Advanced systems will even help the acquisition of expert knowledge by asking experts appropriate questions. A less-experienced user can then exploit the data by applying the stored combined expertise of several experts (e.g., knowledge from experts in assessment, cartography, and real estate law).

The current research in expert systems, expert database systems, and knowledge-based systems seems applicable to LIS. It is recommended that those results be applied and generalized expert system capabilities built into LIS in lieu of trying to achieve essentially the same goals with a combination of very special programs. The formal framework in expert systems allows easy combination and general use of facilities which otherwise have to be constructed several times to apply only in special cases (e.g., usually they provide a powerful, extendible language or a standard user interface).

Recommendations

A query language based on firm mathematical foundations (predicate logic) and with substantial considerations of artificial intelligence results should be built in a transportable form and made available to other research groups and used for prototype applications.

It must be studied how powerful a language is necessary; a balance between the flexibility of extended languages but very complex and lengthy processing requirements and a more limited subset with known fast lengthy processing requirements and a more limited subset with known fast evaluation algorithms must be sought. It is recommended that fast evaluation algorithms must be sought. Horn-clauses as a powerful subset for which fast algorithms are known (Robinson resolution) are also recommended for study.

Organizational Concerns: Data Protection and Distribution

Much work for computerized LIS starts with the assumption that the integration of data is performed by all contributing units bringing data to a central place and then using these collected data jointly. Reality, however, is different. Organizational units may feel strongly about the importance and/or confidentiality of their respective data; this is especially true if cooperation among different agencies, government levels, or private sector organizations is sought.

Depending on specific needs, a unit may permit other units to use some of the data they contribute to the LIS, or permit only reading but no changes by other units. This request should be dealt with by the database management system under the heading "Data Protection" (see Data Protection, *infra*). Sometimes an organizational unit may be very reluctant to transfer its data to a central facility and wish to keep it on a computer under direct control. Such situations can be dealt with by distributed database systems.

It seems very important that for each data element in a database, a responsible organizational unit is determined; it is recommended that only this unit be able to change the data. Methods for reporting encountered inaccuracies from other units must be integrated into the LIS.

Data Protection

A commercial database provides protection of data from unauthorized users. Any DBMS used for an LIS must include such facilities. It must be possible to assure that users not gain access or change. Land information systems differ from commercial information systems, since the description of what data are accessible to a certain user is based not only on entity and attribute types but also on the spatial locations of the reference object (e.g., the assessor of town x may not change assessment records in town y).

Protection of Aggregate Data

Some users may not be authorized to use some data in a one-unit form but permitted to use statistically aggregated data. This is mostly true for confidential data in town administrations and for census data. It has been found that, in general, there are no secure methods to allow users unlimited access to aggregate data without compromising confidentiality. Sophisticated users can retrieve data in answer to several aggregating queries and deduce the values for one single individual (Denning, 1980). It should be studied whether the spatial aggregation which is predominantly used in an LIS is a secure subset of aggregating operations.

Distributed LIS

In computer networks, it is not necessary that every organizational unit maintain its own LIS. It is possible to access data that are stored in another location. In more advanced forms, it is possible to use several databases at different locations as if they were a single central database (distributed databases (Dayal, 1983)).

There are two problems peculiar to LIS.:

1. The data are often distributed based

on spatial location of the reference object (e.g., all assessment records for a town are found in the database of that town, whereas the assessment records of the next town are found in the next town. Generally it may be expected that this form of distribution is simpler and better controlled than the most general form of a distributed database as discussed in the literature (Lampson, 1981).

2. Geometric data are voluminous and transfers from one location to another may be unacceptably slow. There are a number of possible solutions (e.g., faster communication lines, compression of the data, data replication at the using site) which are peculiar to LIS and should be investigated.

Recommendations

It should be studied how the special spatial distribution of responsibility in an LIS affects data protection and distribution. Specifically, the questions of whether spatial aggregation of data preserves confidentiality and when spatially aggregated data may be distributed to third parties are important.

It should be studied from an organizational and a technical point of view how distribution of data and responsibility for data interact and what solutions are adequate.

Data Storage and Retrieval

Building an integrated LIS means storing large amounts of data and retrieving them to produce the answers that users need for their decisionmaking. The concept of database management systems is to concentrate these functions in a single software module. This entails a number of advantages, including the following:

- A DBMS can be written in a general manner, then adapted for many different uses.
- A DBMS provides the only access path to the data and can thus protect the data in various ways.

Various studies have established that the standard DBMS used in commercial applications are not suitable for LIS and other geometrical-oriented applications (Rehfuss, 1985; Frank, 1980). The database community has lately started to study so-called DBMS for nonstandard applications, but it does not seem likely that a single solution can be found.

Object-Oriented Database

A database for an LIS should support an object-oriented approach to software design. The database must let the application programer store and retrieve data describing objects without restrictions on their internal structure or the operations appropriate for them.

Available database systems show severe restrictions in the methods an application can use to map its data for storage and often include a large number of functions which are useful for commercial applications but lack geometric data types and operations. An object-oriented database design may match the needs of an LIS designer better and it is believed that such a DBMS can be formally described and thus made easy to use.

Support for Fast Spatial Access

Land information systems need fast access methods to databases on spatial location of the objects to which the data refer. There are a number of relatively similar systems which have been published lately. It is not clear how much they differ and for what specific application each of them is advantageous.

Multiuser Database

Most commercially available databases allow two or more users to access simultaneously; however the data storage systems used for geographic and land information systems in the past have not permitted this. It seems important for the application of an LIS by multiple organizational units that users be able to access and change data independent from each other and that the database management system prevent any undesirable interactions among users.

Currently it is not known which techniques for controlling multiuser access to databases is most efficient for LIS. The specific structure of geometric data and the respective operations may make some of them more advantageous than others. A study should also consider the relationships among distribution of databases on several machines, the use of intelligent workstations for data presentations, and control of interaction of multiple users.

Long Transactions (Plouffe, 1984)

Traditionally, changes in multiuser databases are grouped in transactions which lead the database from one consistent state to the next. All data affected by a change in a transaction are locked and not accessible for other users. This system is appropriate for the commercial world where changes are small and immediately concluded. In LIS databases (as well as in CAD/CAM systems), some connected changes may take a very long time from initiation until completion. A locking of all referenced data during this period may not be acceptable. (This is especially a problem with geometric data.) On the other hand, the changes need to be wrapped into some kind of a transaction because the database is not consistent while the change is being made.

Recommendations

It is recommended that a study be made to identify the special requirements an LIS has for a DBMS and propose operational solutions. The system must support all the functions the basic query language needs (see section on Integration of User Requirements for Data Retrieval, *supra*) and should blend well with an object-oriented approach to software engineering. The result of the study would be a formal specification of a DBMS for an LIS, including datastructures and corresponding operations.

An investigation into the response time and the organization of physical data storage would be beneficial. This study should include a comparison of currently known methods for fast spatial access, a description of these methods in a uniform format and a comparison of relative merits for specific cases.

The research community would benefit greatly if such a system could be implemented in a transportable language and provided to other groups that are working in the area.

The transaction concept from commercial databases has to be extended to encompass the so-called "long-transactions" in LIS and control consistency constraints and multiuser interaction across such changes.

Presentation of Retrieved Data

Presentation specifications are usually included in the query formulations, and presentation methods are integrated in the query processing packages. It might be advantageous for interactive graphic operations to consider database retrieval and presentation modules jointly. A mode of operation where the database delivers the retrieval data to a presentation module which then produces the output (e.g., a map) seems inefficient because it requires the presentation module to rebuild the datastructure which is already present in the database. Furthermore, the production of map grahics is not so straightforward that all data necessary can be requested at the beginning. A mode of interaction among the different modules should thus be studied.

However, we do not believe that data describing the graphical presentation of data should be included in the database (e.g., placement of names on maps). It has been shown eloquently by Freeman (1984) that such purely graphical descriptions depend very much on the window, scale, and other contents of the map, and are, therefore, of little value in a system which produces maps on demand.

Recommendations

The rules for automatic production of map graphics should be studied, including automatic symbol and name placement and automatic generalization. A first goal of such a software package should be the automatic production of acceptable maps for presentation to the user on screen (soft copy maps). The study should concentrate on the necessary fundamental operations to provide formal models of operation.

Conclusion

Integrating data from different users into an LIS and making these data later available to users pose a considerable challenge. To build computerized systems, one needs a more general approach than for the design of a system with a specific, narrowly defined task. To achieve the necessary flexibility to follow or, better, foresee changing needs, the fundamental building blocks must be identified and methods for their combination found.

Most of the basic requirements for an integrated data storage and retrieval mechanism for an LIS are reasonably well known. Further efforts to make more detailed lists by studying more different user groups are not presently necessary. First, the requirements should be theoretically analyzed and formally described. As far as possible, mathematical methods must be applied and most general solutions attempted. "Systems should not be built for our imperfect technology, but should reflect the inherent structure of the problem." (Chrisman, 1985)

It may be necessary to build working prototypes to prove the feasibility of certain solutions, and it would be beneficial if transportable modules providing some basic functions could be written and distributed to all research teams in this area.

REFERENCES

Aronson, P. (1985), "Applying Software Engineering to a General Purpose Geographic Information System," *Proceedings, AutoCarto 7*, Washington, D.C.

Boyle, A. R. et al. (1983), "Final Report of a Conference on the Review and Synthesis of Problems and Directions for Large Scale Geographic Information System Development," Apr., NASA Contract NAS2-11346; avail. from ESRI, Redlands, Calif.

Borgida, A., J. Mylopoulos, and H. K. T. Wong (1984), "Generalization/Specialization as a Basis for Software Specifications," *On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages*, M. L. Brodie et al. (eds.), N.Y.: Springer Verlag.

Chen, P. (1976), "The Entity-Relationship Model: Towards a Unified View of Data," *ACM TODS*, Vol. 1, No. 1.

Chevallier, J.-J (1984), "System Approach to LIS: A Means to Coordinate Research and Development in the Area of LIS," *The Decision Maker and Land Information Systems*, A. C. Hamilton and J. D. McLaughlin (eds.), Proceedings, FIG International Symposium, Edmonton, Alberta, Oct.

Chrisman, N. (1985), "Alternative Routes to a Multipurpose Cadastre: Merging Institutional and Technical Reasoning," *Proceedings, AutoCarto 7*, Washington, D.C.

Codd, E. F. (1982), "Relational Database: A Practical Foundation for Productivity," *Comm. of the ACM*, Vol. 25, No. 2, Feb.

Dayal, U. (1983), "Processing Queries Over Generalization Hierarchies in a Multidatabase System," *Proceedings, Ninth International Conf. on Very Large Databases*, M. Schkolnick and C. Thanos (eds.), Florence, Italy, Oct.

Denning, D. (1980), "Secure Statistical Databases with Random Sample Queries," *ACM Transactions on Database Systems*. Vol. 5, No. 3, Sept.

Findler, N. V. (ed.) (1979), *Associative Networks—Representation and Use of Knowledge by Computers*, N.Y.: Academic Press

Finin, T. and D. Silverman (1984), "Interactive Classifications of Conceptual Knowledge," *Proceedings, First International Workshop on Expert Database Systems*, L. Kerschberg (ed.), Kiawah Island, Oct.

Forrest, R. (guest ed.) (1984), "Introduction," *ACM-Transactions in Graphics*, Special Issue on Computational Geometry, Vol. 3, No. 4, Oct.

Frank, A. (1982), "Application of DBMS to Land Information Systems," *Proceedings, 7th International Conference on Very Large Databases*, C. Zaniolo and C. Deloebel (eds.), Cannes, France, Sept.

Frank, A. (1982), "Data Structures for Land Information Systems—Semantical, Topological and Spatial Relations in Data of Geo-Sciences," Ph.D. thesis, Swiss Federal Inst. of Tech., Zurich, Switzerland.

Frank, A. (1981), "Land Information Systems—Theoretical and Practical Problems," Paper, 305.1, Federation Internationale des Geometres (F.I.G.), XVI Congress, Montreux, Switzerland.

Frank, A. (1982), "Conceptual Framework for Land Information Systems—A First Approach," paper presented at 1982 meeting of Commission 3 (LIS), F.I.G., Rome, Italy; Univ. of Maine, Surveying Program, Rept. 38.

Frank, A. (1984), "Requirements for Database Systems Suitable to Manage Large Spatial Databases," *Proceedings of the International Symposium on Spatial Data Handling*, D. Marble, et al. (eds.), Zurich, Switzerland.

Franklin, W. M. (1984), "Cartographic Errors Symptomatic of Underlying Algebra Problems," *Proceedings of the International Symposium on Spatial Data Handling*, D. Marble et al. (eds.), Zurich, Switzerland.

Freeman, D. and J. Ahn (1984), "AUTONAP—An Expert System for Automatic Name Placement, *Proceedings of the International Symposium on Spatial Data Handling*, D. Marble et al. (eds.), Zurich, Switzerland.

Gougen, J. A. (1978), "An Initial Algebra Approach to the Specification, Correctness, and Implementation of Abstract Data Types, *Current Trends in Programming Methodology*, R. T. Yeh (ed.), N.Y.: Prentice Hall, *Data Structuring*, Vol. 4.

Guttag, J. (1977), "Abstract Data Types and the Development of Data Structures," *Comm. of the ACM*, June.

Isner, J. F. (1982), "A Fortran Programming Methodology Based on Data Abstraction," *Comm. of the ACM*, Vol. 25, No. 10, Oct., p. 686.

Kent, W. (1981), "Data Model Theory Meets a Practical Application," *Proceedings, 7th International Conference on Very Large Databases*, C. Zaniolo and C. Diobel (eds.), Cannes, France.

Kerschberg, L. (ed.) (1984), *Proceedings of the First International Workshop on Expert Database Systems*, Kiawah Island, S.C., Oct.

Lampson, B. W. et al. (1981), *Distributed Systems—Architecture and Implementation, An Advanced Course, Lecture Notes in Computer Science 105*, Berlin: Springer Verlag.

Mead, D. A. (1981), "Statewide Natural-Resource Information Systems—A Status Report, *Jour. of Forestry*, Vol. 79, pp. 369-372.

Naisbitt, J. (1984), *Megatrends*, N.Y.: Warner Books.

National Research Council, Panel on a Multipurpose Cadastre, Committee on Geodesy (1980), *Need for a Multipurpose Cadastre*, Washington, D.C.: National Academy Press.

Palmer, B. (1984), "Symbolic Feature Analysis and Expert Systems," *Proceedings of the International Symposium on Spatial Data Handling*, D. Marble et al. (eds.), Zurich, Switzerland.

Parnas, D. L. (1977), "A Technique for Software Module Specifications With Examples," *Comm. of the ACM*, Vol. 15, No. 5, May.

Plouffe, W. et al. (1984), "A Database System System for Engineering Design, *IEEE Bulletin on Database Engineering*, Vol. 7, No. 2, June.

Rescher, Urquhart (1971), *Temporal Logic*, Berlin: Springer Verlag.

Rehfuss, S. et al. (1985), "Particularly in Engineering Data," Kerschberg, L., *op. cit.*

Robinson, V. and A. Frank (1985), "About Different Kinds of Uncertainty in Collections of Spatial Data," *Proceedings, AutoCarto 7*, Washington, D.C.

White, M. S. (1984), "Technical Requirements and Standards for a Multipurpose Geographic System," *The American Cartographer*, Vol. 11, No. 1, Apr.

Wirth, N. (1983), *Programming in Modula-2*, 2d ed., Berlin: Springer Verlag.

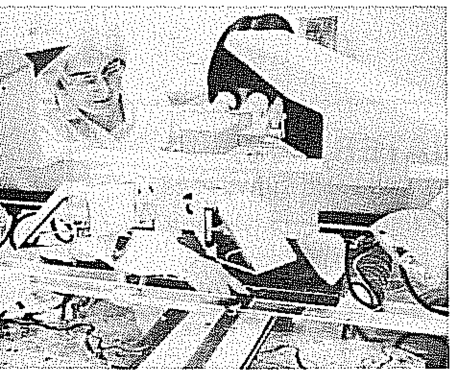
Zilles, S. N. (1984), "Types, Algebras and Modelling," *Conceptual Modelling, Perspective from Artificial Intelligence, Databases and Programming Languages*, N.Y.: Springer Verlag. ■

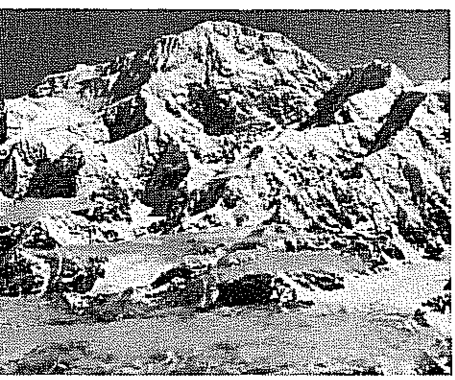
–1986 ASPRS-ACSM FALL CONVENTION

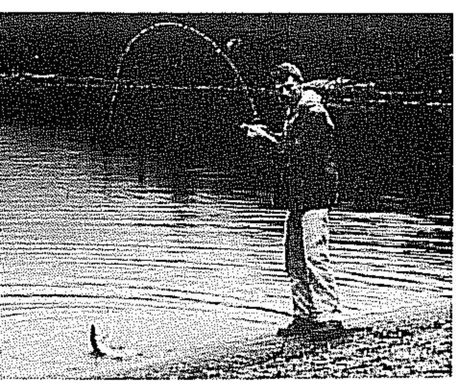
New Frontiers

Come to Alaska

ew Frontier Ideas
.ast Frontier Fun







Anchorage, Alaska
September 28 - October 3

NEW FRONTIERS OF TECHNOLOGY

Convention Center full of Exhibits of technology. Five days of technical papers focusing on:

- High latitude surveying and mapping.
- Multipurpose Cadastre
- Geographic Information Systems
- Automated Cartography
- Digital Processing
- Surveyor and the Law
- Emerging Technology
- Photogrammetry and Remote Sensing
  - —Land and Ocean Applications
  - —New techniques and tools
  - —Resource Management

FRONTIER FUN

Alaska salmon bake in an airplane hangar.
Alaska history in story and song at the Banquet.
Once-in-a-lifetime hunting or fishing trip.
Alaska friendliness and hospitality.
Cruise the Inside Passage on your way up.
Ride the train to Denali National Park.

SEE THE FRONTIER

Unspoiled wilderness of the Great Land.
Majestically beautiful scenery.
Awesome mountains and Glaciers.
Pioneer gold camps.
Colorful Native villages.
Glorious fall color against snow-capped peaks.
Mount McKinley, North America's highest mountain.

OFFICIAL CONVENTION COORDINATOR
TRAVEL ENTERPRISES
517 W. Northern Lights Blvd.
Anchorage, Alaska 99503
800-544-2207

The American Society for Photogrammetry and Remote Sensing
and
The American Congress on Surveying and Mapping

1986 ASPRS-ACSM Fall Convention
P.O. Box 103899
Anchorage, Alaska 99510