MANAGEMENT OF SPATIALLY REFERENCED DATA

Andre Frank
Institute of Geodesy and Photogrammetry
Swiss Federal Institute of Technology (ETH)
Zürich, Switzerland

Markku Tamminen
Laboratory of Information Processing Science
Helsinki University of Technology
Espoo, Finland

16

ABSTRACT

Management of spatially referenced (geo-) data forms the basis of land information systems. To build open-ended, reliable and efficient systems we must understand what geo-data is and what special requirements have been posed by its management. Here we provide a framework for conceptual definition.

The management of spatially referenced data is approached from an object-oriented viewpoint arriving at a conceptual definition of data contents, often called a data model. From integrity and access requirements a general interface to geo-data and a layered model for implementing specific access functions and geometric structures is obtained. Efficiency considerations and the role of general data base management systems are also addressed.

INTRODUCTION

Land Information Systems (LIS)

Information systems are vital for modern organizations to manage their physical and abstract assets. Land and related resources become scarce. Therefore, society needs adequate information for its management. The concept of a Land Information System (LIS) has been proposed to coherently bind together these important data and their management. It embodies much of the traditional function of the surveying profession together with many related activities.

The following definition was adopted by the International Federation of Surveyors at its meeting in 1981:

> "A Land Information System is a tool for legal, administrative and economic decision-making and an aid for planning and development, which consists, on the one hand, of a data base containing spatially referenced land-related data for a defined area and, on the other hand, of procedures and techniques for the systematic collection, updating, processing and distribution of the data.
>
> The base of a Land Information System is a uniform spatial referencing system for the data in the system, which also facilitates the linking of data within the system with other land-related data." (FIG, 1981)

Characteristics and Problems of Spatially Referenced Data

This paper deals with data related to spatial objects. The data may describe the position or other geometric properties of an object, and may also contain other information. An example is a parcel, for which we may store its position, geometric data of its boundaries, the name of its owner, tax-value, land use, etc.

An object is called spatially referenced when its data determines its geographic location. The set of object data is also called spatially referenced or 'geo-data' and, in general, the prefix 'geo-' is used in a corresponding meaning.

General data base management systems (DBMS) have been developed as tools to manage administrative data bases. What is needed is the ability to use "general geometric knowledge" both for defining the data (data contents, rules of integrity) and manipulating it efficiently (e.g. retrieving objects close to a certain point).

Without proper consideration for data definition, information in a collected set of data may be incomplete, making it worthless for some applications. This problem is common to most large data systems, but it is more difficult to avoid in a LIS because of the traditionally non-numerical (graphical) nature of spatially referenced data collections (i.e. maps). Existing systems often show deficiencies, due to two causes.

The most common error is defining geographical objects solely by the graphical appearance of some derived output (i.e. map). This geometry oriented approach is supported by so-called interactive graphical systems (GIS) which have attained a high level of user-friendliness and efficiency in interaction, producing impressive results (Kevany, 1979). However, the cost of efficiency has often been incomplete data contents; all input objects and relationships may not be automatically reconstitutable from the data base. Also, these systems often rely solely on interactive operation, neglecting the possibilities of automatic inference. This is serious when data integrity is affected.

A Land Information System may also be built just as an ordinary administrative data processing application utilizing a general data base management system. Some integrated systems have been designed and built in this way (SEMA, 1979; ICOREM, 1978; IBM, 1979). There are no theoretical problems in this approach; the data base contents may easily be defined to be complete and the DBMS provides for good data protection. However, the systems are often large and expensive, yet offering a rather low level of automation due to the lack of consideration given to geometric methods.

Conclusions for This Research

It is somewhat surprising that no minimal set of concepts for geo-data seems yet to have found general acceptance although some proposals have been made (Burton, 1979; Cox, 1980). For geo-data, we need ordinary data base management facilities (augmented by methods for treating geometric data).

End users and programmers should be given concepts and tools resembling those of ordinary human inference on the location and form of objects. Data base designers should be able to define geographical data "as it is", detached from implementation considerations. Finally, data base administrators should have tools to automatically guard the very valuable asset formed by the data. The data base approach is a sound foundation for this. However, a DBMS must be complemented with a layer containing general concepts of geometry and geo-data. The result may be called a geo-data management system (GDMS).

Compared to traditional data base management systems, a GDMS is aimed at facilitating automatic geometric inference and integrity and the use of geometric concepts in connection with a fixed spatial reference for objects. By "geometric inference", we mean relationships derivable from the geometry, and "natural geometric concepts" means pointing, intersection, inclusion, vicinity, etc. Fixed spatial reference means that an object is situated at some well defined location. These concepts and tools will lead to enhanced efficiency in the construction of systems and data bases, and thus to a higher level of integration and data integrity.

Efficiency is crucial because the task of collecting data or transforming old graphical maps and other archival data into digital form is huge. It easily costs tens of millions of dollars for a city, and several orders of magnitude more for a whole country. The data base sizes may be of the order of 10^3 and 10^5 million bytes respectively.

This article is based on research carried out separately since 1977 at the Swiss Federal Institute of Technology, Zurich, and since 1978 at the Data Center of Helsinki Metropolitan Area and the Technical University of Helsinki.

DATA CONTENT DESCRIPTION

Background

Traditional maps present geo-data efficiently for human inference. Their high information contents are achieved by not requiring explicit identifiers to describe spatial relationships. Systems that cannot utilize implicit spatial relationships and identification are bound to be less user-friendly and efficient (Weber, 1978; Erstling, 1979; IBM, 1979).

It is recognized that a so-called object oriented approach is necessary. The view to data is from objects to geometry: "Object O consists of lines $L_1, \ldots, L_n$". Nevertheless, spatial inference often requires the basic geometric components of objects to be considered as primary. In this geometric approach the view to data is bottom-up: "Line L belongs to objects $O_1, \ldots, O_n$". Here it is imperative to recognize when inaccurately localized geometric components of objects refer to the same points or lines.

Geo-data management should support both the above views. The first one is the most natural for defining information systems, while the second one is primary for preserving geometric data integrity and for efficiency in many processing tasks (e.g. map production).

Traditionally, data base management is concerned with entities described by sets of attributes and explicitly defined relationships between the entities. Another way to describe data is by abstract data types: we specify a set of entity classes (data types) together with a set of allowed operations. Both description methods are useful for geo-data management.

Definitions of Concepts Used

Objects are any real-life entities treated in a geo-data application. In a cadastral system, objects could be buildings, parcels, blocks, etc., or in an underground network information system, they could be streets, cable lines and supporting construction.

The geometric aspects of objects are described by points, lines and areas. Without loss of generality we consider nodes and lines the primitive geometric units. A node is a point given special meaning (e.g. the junction of three lines).

Examples of lines are:

- a straight line segment
- a polygonal lines with a variable number of points
- an arc of a circle defined by three points.

Areas, which are defined by their boundary lines, are more complex geometric units.

Objects can be classified on the basis of their data contents (the kinds of attributes a certain class of objects may possess) or on the basis of their geometric structures (what geometric properties can be assumed for an object or set of interrelated objects). For example, utility lines for water, electricity or gas have a similar geometry (Figure 1) as opposed to parcels, census tracts, or county subdivisions (Figure 2).

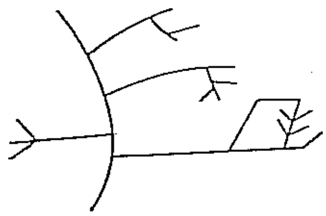


Figure 1: Network

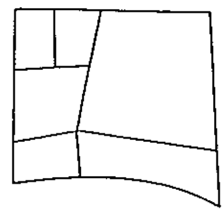


Figure 2: Partition (Polygon)

We call such abstract types of geometry, together with the operations allowed on them, Basic Geometric Structures. For example, the geometry of Figure 2 may be described as a "complete cover" partition (polygon) and it is reasonable to ask questions such as "Which polygon contains point (x,y)?". The same question is not meaningful for geometries of the type shown in Figure 1.

Abstract geometry denotes the geometry "meant by" a possibly inprecise observed geometry. Abstract geometry is an informal concept because, by observing only the data gathered in the field, we cannot know the true geometry meant by it. However, we assume that to each observed primitive geometric unit there corresponds a certain unique "true" geometric entity.

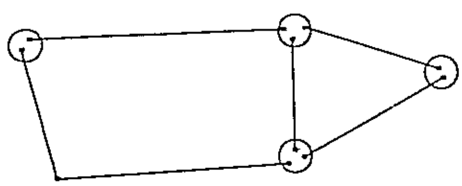


Figure 3: Node uniqueness. All observed nodes within a circle mean the same abstract geometric point.

An observed geometry is called unambiguous if each primitive geometric unit of the abstract geometry corresponds to exactly one primitive geometric unit in the data base. For nodes, this condition is called node uniqueness (Figure 3).

The purpose of a Geo-Data Management System (GDMS) is to manage data describing object geometries and related attributes facilitating the evaluation of geometric relationships. Especially important geometric relationships are the so-called elementary topological relationships (Figure 4), which are determined by lines sharing a common node or objects sharing a common primitive geometric unit.

The geometry of a planar object can be described by a set of lines and nodes. In abstract geometry, all direct neighborhood relationships are determined by objects owning common primitive geometric units. The geometric relationships between objects can be derived from this set assuming knowledge of the corresponding basic geometric structure. This "part-of" relationship suffices to completely describe the geometric contents of a data base and forms the foundation of the data model in the following paragraph. If the observed geometry is unambiguous, all elementary topological relationships can be correctly derived from the version of the geometry stored in the data base.

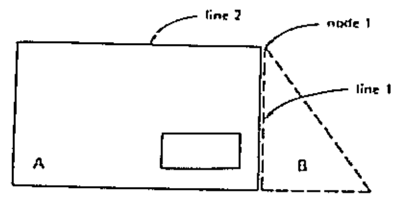


Figure 4: Object A (continuous line) and B (dashed line) are elementary topological neighbors by having line 1 in common. Lines 1 and 2 are also elementary topological neighbors at node 1.

Data Model

A data model is a set of concepts and a related formalism which describes a part of the real world. Traditionally, data models are related to the static aspects (data contents) of data bases. Various data models given in the literature can be used to describe a data base or, as here, the data of an application area. We have chosen the entity relationship diagram (Ullman, 1980) for the most general definition of geographical data (Figure 5).

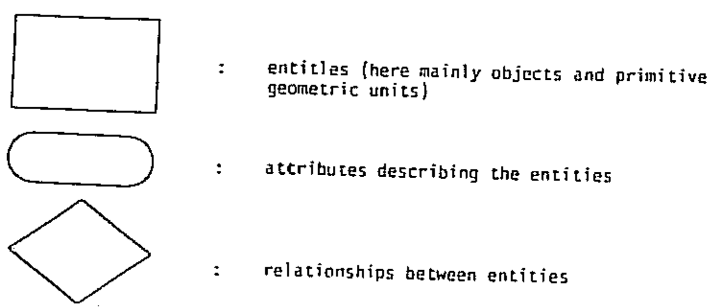
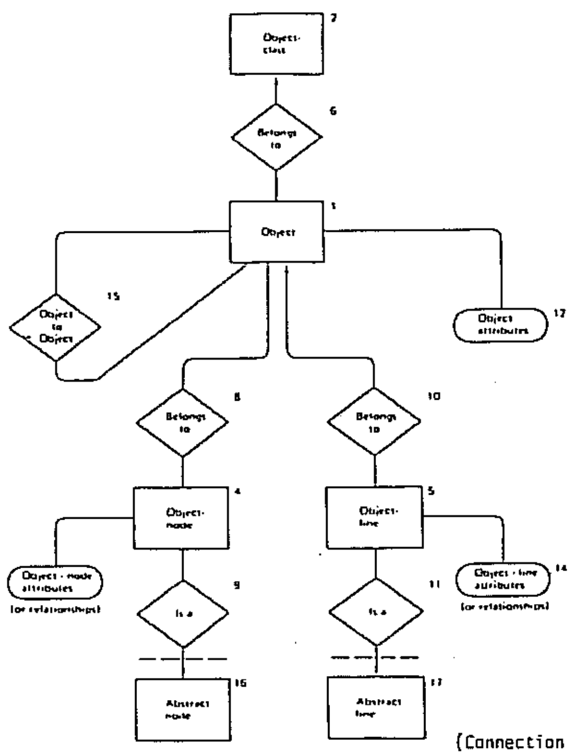


Figure 5: Basic Concepts of the Entity Relationship Diagram

We present a general entity relationship diagram for geo-data in two interconnected figures (Figures 6 and 7) corresponding to the object and geometry oriented view mentioned above. Because we are not considering a single data base, we must use the symbolism of Figure 5 for generic classes of entities, attributes and relationships.

In this form the data model helps us define the relevant aspects of geo-data contents in general for the rest of this paper. It is applied to specific data bases by replacing each generic entity by one or more entities having homogeneous data contents, and by deleting unnecessary parts (Figure 8).



(Connection to Figure 7)

Figure 6: Entity Relationship Diagram of the Object Oriented View to Geometry.

Explanations:

1. Object: A primitive entity of the application.

2. Object class: A classification of objects determining all aspects relevant to the data description.

4. Object-node: A node considered as part of an object.

5. Object-line: A line considered as part of an object.

6. Belongs to: An object belongs to an object class.

8. Belongs to: An object-node belongs directly to the object.

9. Is a: An object-node means an abstract node.

10. Belongs to: An object-line belongs to the object.

11. Is a: An object line means an abstract line.

12. Object attributes: Attributes functionally determined by "object" (e.g. object identifier).

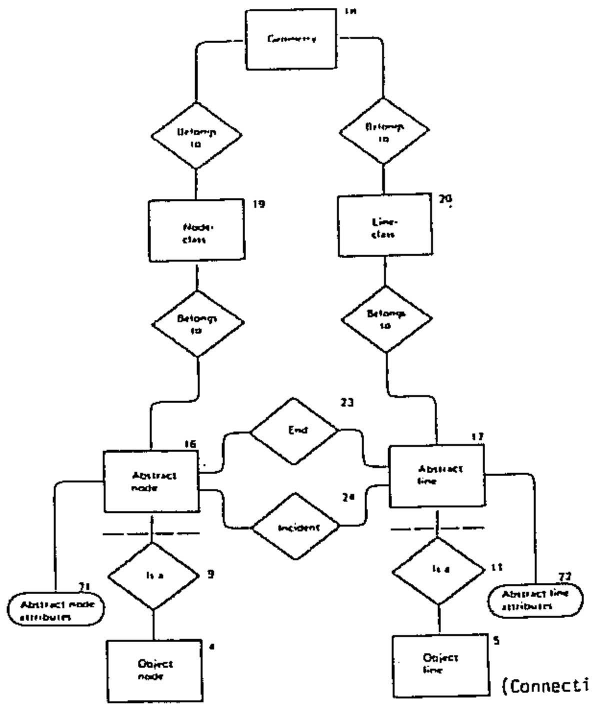
13. Object-mode attributes: Attributes functionally determined by "object-node".

14. Object-line attributes: Attributes functionally determined by "object-line" (e.g. an attribute indicating whether the line is a right/left/center line of a road).

15. Object relationships: Non-geometric relationships between objects, not considered further here.

16. Abstract node: See Figure 7.

17. Abstract line: See Figure 7.



(Connection to Figure 6)

Figure 7: Entity-Relationship Diagram of the Abstract Geometry View to Point and Line Geometry

Explanations:

4.,5. Object node, Object line: See Figure 6.

8.,9. Is a: See Figure 6.

16. Abstract node: A geometric point which has been named a node.

17. Abstract line: A geometric line (a one dimensional point set).

18. Geometry: A collection of nodes and line.

19. Node class: A classification of nodes determining all aspects relevant to the data description.

20. Line class: A classification of lines determining all relevant data description aspects and geometric type (e.g. straight line or line with internal points).

21. Abstract node attributes: Attributes functionally determined by "abstract node" (e.g. precision, coordinates).

22. Abstract line attributes: Attributes functionally determined by "abstract line" (e.g. precision or parametric definition of the line.

23. End: An abstract line is specified to have two end nodes.

24. Incident: An abstract node may have many lines ending (incident) at it.

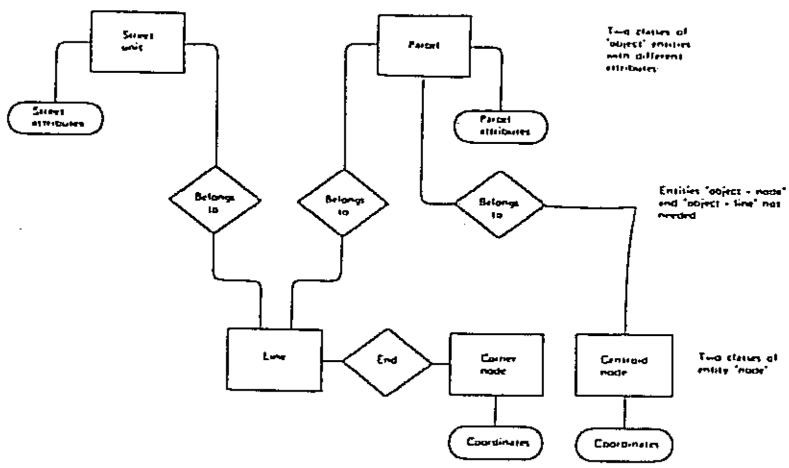


Figure 8: An example of applying the data model to a specific cadastral application, where street units and parcels have common straight border lines and together form a partition (complete cover).

Figures 6 and 7 help define the two main approaches to geo-data management. In the object oriented data bases. the data description often contains only Figure 6. Such data bases are simple to implement, because recognition of primitive geometric units, common to several objects, is not directly addressed. The main requirement is that lines be efficiently representable as a primitive data type and that geometric operations (e.g. intersection) can be used (Cox, 1980). Geometric integrity is the responsibility of the users.

The relevance of the classification of objects to object classes has been established by cartographic practice (i.e. in defining which real-world objects are presented by which cartographic notation). In various countries (Brüggemann, 1979; ISOK, 1980; USGS, 1980; AS, 1981) an effort has been made to standardize cartographic file transfer (including the meaning of the data) by establishing standard transfer file formats and object code catalogues classifying objects.

These code-systems, because they have been conceived in the context of the geometry oriented approach and are intended for data transfer as sequential files, are limited in expressing relationships. Often they do not clearly separate encoding of attributes related to conceptually different data types. For instance, the code attribute of a line in such a system may simultaneously refer to an object, indicate an object-to-object relationship and line attributes. A data model with more structure, such as that of Figures 6 and 7, should be used for defining geo-data and, as a corollary, geo-data exchange standards.

The data model gives a means of describing geographical entities and their attributes at a conceptual level separated from implementation aspects. It enforces a distinction of objects (application specific entities) from geometry (more universal), and produces a classification of attributes by so-called functional dependencies (Ullman, 1980). This aids in avoiding the ambiguity that may be caused by attaching attributes directly to geometric units.

The data model is minimal; classifying general attributes of an binary relationships between nodes, lines and objects requires at least this number of concepts. However, it is, in a way, also complete; the abstract geometry and attributes of any collection of objects may be described by it.

Implementing Geo-Data Systems using Available Generalized Data Base Management Systems

For most computers, software for data base management is available. Such systems are based on different data models: hierarchical, network or relational.

The logical data model elaborated on in Figures 6 and 7 can be easily adapted to the network or relational data model. Available software can and probably should be used in these cases.

In order to implement a geo-data system, a number of representational details have to be fixed. These decisions influence mainly performance but this is often crucial to the success of a project. There are sometimes alternatives for storage; instead of storing a data value, it may be computed from the values of other data on each access. For example, the surface area of a parcel can be stored or recomputed each time the user wishes to know it. A similar decision has to be made regarding access paths. It makes an enormous practical difference if we find a data element within a couple of disk accesses (response time 1 second) or if we have to search sequentially the whole data base (response time some hours perhaps).

Each relationship in Figures 6 and 7 is a potential access path. Implementation of a conceptual data model consists largely of choosing the best representations (implicit or explicit) for each relationship.

REQUIREMENTS FOR A GEO-DATA BASE MANAGEMENT SYSTEM

Integrity Constraints

An integrity constraint is a logical proposition that the data base contents must satisfy. Geometric integrity constraints are perhaps the most typical and demanding aspect of geo-data management. These are geometric propositions, e.g. "No lines belonging to a certain object type meet except at end nodes", forming one of the main ways of enhancing data quality. They also largely determine the internal access requirements of a geo-data management.

Structural integrity constraints describe a certain basic geometric structure and should hold for any observed or abstract geometry known to be of that type, irrespective of the actual coordinate values measured. For instance, an area must have a closing border. The purpose of empirical integrity constraints is to define when we mean the same abstract primitive geometric unity for two observed ones. For instance, two nodes may be taken to mean a single one if they are too close to each other.

In a geographic data base, structural integrity constraints should override the observed geometry. As a result, objects that are known to satisfy the constraints of a certain geometric structure may always be retrieved by access functions specific to that structure. A main purpose of geometric integrity constraints is to make elementary topology derivable from geometry.

Figure 9 illustrates some typical integrity constraints, which are enforced by empirical parameters and structural invariants (e.g. the requirement that a line in an area partition be the border of two areas).

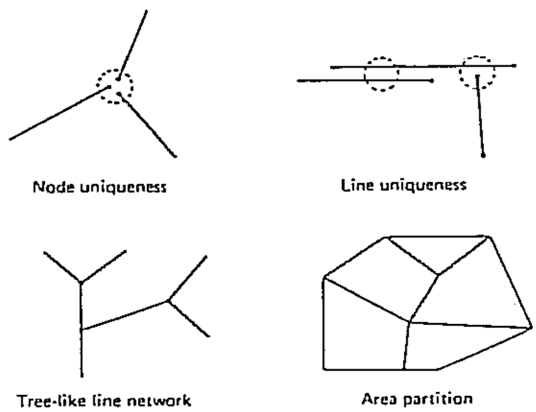


Figure 9: Geometrical Integrity Constraints

A geometric integrity check between an object and the data base is usually composed of checks on each primitive geometric unit of the object. Each partial check can be described as follows.

Procedure 'Geometric Integrity Check'

1. Retrieve from the data base all primitive geometric units relevant to judging the integrity of the new one, e.g. retrieve all nodes within a certain distance from a node to be inserted.

2. If the new primitive geometric unit is certainly acceptable, authorize the intended operation. If it is incorrect, inhibit the operation.

Thus, the (spatial) query for relevant primitive geometric units is a central part of an integrity constraint. This largely determines the access functions and structure of the system. It is shown that integrity checking may make insertion in a geo-data base two orders of magnitude more expensive than retrieval (IBM, 1979).

Geographical Data Independence

The pursuit of data independence is, perhaps, the main driving force in data base research. Data independence means the separation of conceptual contents from implementation details. Programs should not need to be changed when the representation of data is changed or new types of data are brought in.

Geometric integrity constraints (e.g. node uniqueness) should be maintained by the system automatically. A tentative informal definition of geographical data independence summarizes this.

A geo-data base is data independent if:

1. It satisfies the ordinary criteria of data independence by allowing attribute representation and internal access paths between entities to be changed without affecting application programs.

2. It supplies a "small", well-defined set of external functions through which all applications access geometric data.

3. The external access functions are independent of the internal method of object retrieval and are capable of accessing an object as a whole.

4. The external access functions are independent of any auxiliary data base units (e.g. cells) or auxiliary identifiers (e.g. line numbers).

5. Geometric integrity constraints are, as far as possible, defined below the external interface.

Geographical data independence leads naturally to the concept of geo-data types. The three abstract data types defined by Cox (ie. point, line and area) do not, however, seem sufficient from the point of view of integrity.

ACCESS TO GEO-DATA

A Layer Model

The different functions within a geo-data base management system should be grouped into layers. The functions in each layer should use the services of the functions of the underlying layers and each layer should form a reasonable abstraction of a part of the whole task.

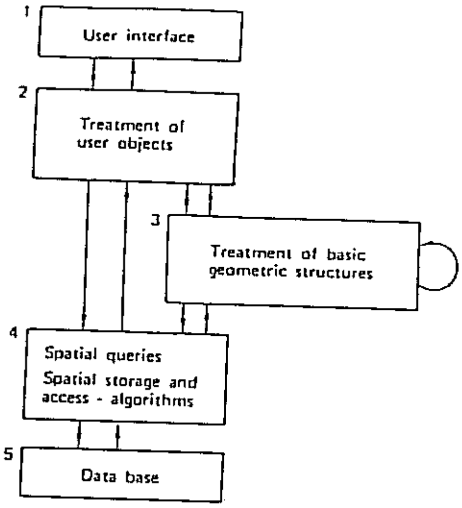


Figure 10: Part of a Layer Model for Geographical Data Management

Figure 10 describes the layer structure that we recommend for a GDMS (Frank, 1981a).

1. The user accesses the data base either using a specialized application interface ("Push button x to register a new parcel") or a general geo-query language defining spatial queries ("Retrieve all object type-list within window") (Frank, 1982).

2. The treatment of application specific objects containing a geometry plus other attributes and relationships is accomplished by a specific layer. Each object class is implemented as an abstract data type defined by its operations. Part of this is ordinary application programming, while the most difficult part is supplied by access functions and integrity constraints of the lower layers.

3. Each basic geometric structure is implemented as an abstract data type by defining the allowed operations (including integrity constraints). A basic geometric structure may be implemented using the operations of another, more primitive one. The lowest level implementation consists of spatial access to sets of lines and nodes in the data base. Also, integrity constraints are based on similar access capabilities.

4. Spatial storage and access algorithms implement (efficiently) access to a defined geometric (spatial queries) or topological ("give neighbor area") vicinity. A spatial query is a retrieval operation of the type "Find all data base objects whose geometry intersects the point set determined by a window". They supply the most natural and general interface and basis for structuring geo-data. Depending on the DBMS available, the implementation

of spatial queries relies either directly on set-retrievel (relational DBMS) or on navigation according to the explicit relationships stored in the data base (CODASYL DBMS).

5. The data base is physically organized as low-level data representations, such as records and links between them either using a general DBMS or a tailor-made file system. Only at this level should these general (application independent) data processing concepts be reached.

The above layer model achieves geographical data independently by localizing all DBMS-specific functions in the implementation of Layer 4 and all machine-dependent aspects in Layer 5. The approach has been shown feasible and efficient enough by implementing it on a commercial available CODASYL DBMS (Frank, 1981b).

User Interface

The user may converse with a Land Information System in different ways:

For an application where the interaction between the user and the LIS can be standardized, an application programmer will prepare a procedure simple to use. Example: Retrieval of the owner of a parcel.

In some applications, for instance in town planning, the information needs of the user are difficult to know in advance. There the user is given a query language which permits him to state questions in a form understandable to the computer.

Example: "List all buildings with more than five floors in the Sunny Hill area" could be formulated as (Frank, 1982):

```
LIST

All buildings (GIVE street-name, house-number)

     WHERE number-of-floors GREATER 5

OF

THE area WHERE name = 'Sunny Hill'.
```

Treatment of User Objects

If an object-type is declared to have a geometry of a certain basic geometric structure, it inherits all the operations of the corresponding abstract data type.

The application programmer of Layer 2 should, therefore, only need to learn that a certain object has a geometry as depicted in Figure 1 or 2 and not be concerned about the relatively complex rules regarding integrity constraints for operations on this geometry, all operations being already abstractly defined in Layer 3. The sole concern is the meaningful application of specific operations.

Basic Geometric Structures

The set of basic geometric structures of a geo-data base management system should be open-ended so that the new ones may be defined as needed. We have, on purpose, separated the concept of a basic geometric structure from that of an object:

- A basic geometric structure incorporates geometric aspects only.
- The object refers to the geometry, but contains also attribute values.

By this separation, basic geometric structures may be made to embody "universal geometric reasoning". We do not try to present an exhaustive set of operations on basic geometric structures, but list some typical ones in Table 1.

Table 1: Some Typical Operations of Basic Geometric Structures

Basic Geometric Units	Operations	Description
line-network	check-line	Checks whether a new line-segment may be inserted (e.g. whether it is connected to the network).
	snap-line	Modifies the end nodes of a line if neccessary to connect it to the network.
area-partition	divide	An area in a partition is divided into two separate areas.
	unite	Two adjacent areas of a partition are joined; the superfluous border-line may be dropped.
	get-area	Retrieves the area in which a given point lies.
	neighbor	Retrieves the left or right neighbor area of a line.

The following is an example of using this concept for input of buildings. We admit a general rule embodied in the basic geometric unit, i.e. that no building may cross a border-line of a parcel.

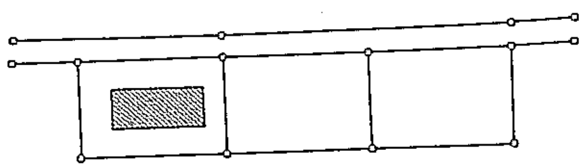


Figure 11: Street, Parcel and Building

This rule may be implemented with a spatial query "Retrieve all parcel border lines within the area given by the building". If there is no such line, the building may be stored. If there is such a line, the building must be rejected because it would violate the admitted rule.

We have hardly touched on operators for the most primitive data types, lines and areas. A comprehensive discussion of these is contained in (Burton, 1979; Cox, 1980).

The efficient implementation of a GDMS requires that the data representation mirror, as closely as possible, the main access type. The structure of the data base should, therefore, correspond as closely as possible to the geographical structure (i.e. location on the earth's surface) of the objects represented. Because geographical space is two-dimensional, but data representations are one-dimensional (basically bit-strings), this is not a trivial problem.

Main Access Type

The access method which is most important in a geo-data system is based on the spatial location of objects. The best example is a query to retrieve all objects of certain types in order to draw a map on the screen of a display terminal. Such a query is of the form:

"Show all (houses, streets, parcels)

within window

(512 < x < 614,

414 < y < 478):"

This type of spatial query can be used both by the user at his terminal and in internal routines of a geo-data system; an example was given in the algorithm in the previous section.

Our solution consists of two parts: A spatial directory logically connects objects and primitive geometric units to geographical access units. These are usually taken to form a uniform or varying grid of cells (Figure 12) with the logical connections described by Figure 13. The spatial directory helps limit the search for objects satisfying a spatial query to those that intersect the same cells as the query window.

Figure 12: Cells of a Spatial Directory

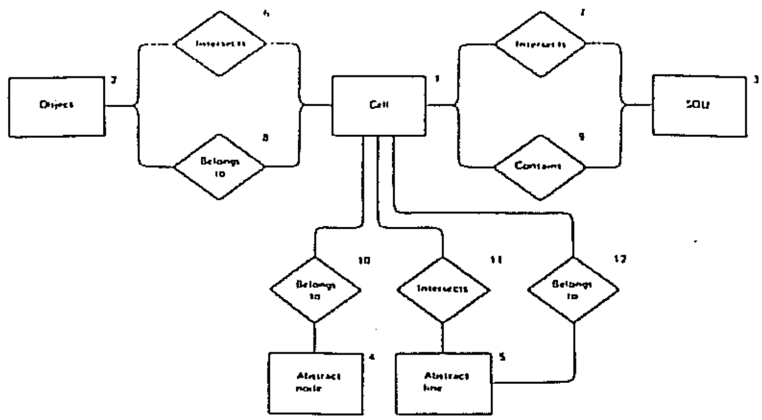


Figure 13: Entity-Relationship Diagram of Geo-Data as Related to the Auxiliary Grid Cell Concept. This concept is often needed because there may be no other natural (or efficient) method of aggregating primitive geometric units for storage and access purposes.

1. Cells: Arbitrary geometric areas (usually rectangles) covering together the geographical area considered.

2. Object: See Figure 6.

3. SQU: Spatial query unit (window), a free area definition (i.e. not necessarily contained in the data base) utilized in formulating spatial queries.

4.,5. Abstract node, abstract line: See Figure 7.

6.,7. Intersects: An object or a window geometrically intersects one or more cells.

8. Belongs to: The geometry of an object may/is required to lie completely within a cell.

9. Contains: When a window totally contains a cell, this may lead to a special relationship.

10. Belongs to: If the cells form a partition, nodes belong to unique cells.

11. Intersects: An abstract line geometrically intersects one or more cells.

12. Belongs to: An abstract line may be completely included in a cell; this is often even required and attained by dividing lines into cell-intersection units (11 and 12 are alternatives).

Due to the characteristics of present-day external storage media, a spatial directory does not suffice for satisfactory performance.

Mass storage systems for data bases have average access times of 30 milliseconds. In order to retrieve over 1000 records in less than 30 seconds (a typical query), each access to the disk should not only retrieve one but many records. As the typical query is based on placement in space (vicinity), records should be placed on the disk in a manner corresponding to the vicinity of the objects represented (Figure 14).

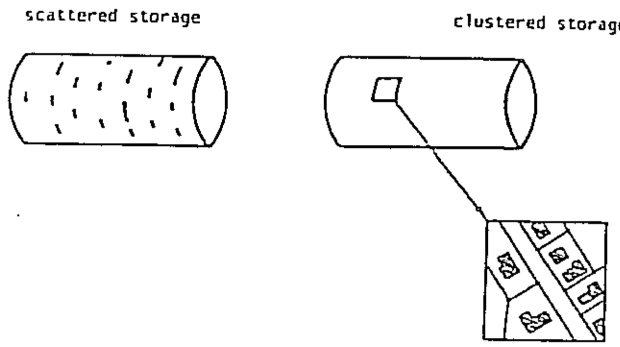


Figure 14: Locality of Access

The above two concepts suffice for efficient access independently of the implementation method. A GDMS cannot be efficiently implemented on a DBMS that does not offer tools to direct (approximately) the physical placement of data. The results are encouraging and show that a transparent structure with simple access algorithms may be achieved. The structures adjust to the configuration of objects without requiring the user to explicitly define the cells or refer to them in queries. This is one component of geographical data independence of the implementation either on a CODASYL DBTG type DBMS (Frank, 1981a) or a relational DBMS or a tailor-made file system (Tamminen, 1981).

PRACTICAL APPLICATIONS

An Existing Land Information System

Several existing municipal information systems integrate a complete geometric description of parcels, streets, etc., with administrative data. Besides automatic production of a unified series of maps, such systems help achieve maximum consistancy and automation in surveying and cadastral functions (Kevany, 1979).

Most of the integrated systems use a general data base management system at least for the administrative data. Some include an interactive graphical system as a more or less self-contained component.

As an example, we present a system typical of the most advanced. Among existing systems it has achieved a very high level of integration and has been successfully used daily for several years (Messmer, 1977). (This is not very common; many ambitious projects have failed.) Other interesting systems are reported in (UDMS, 1981).

Scope of the system:

Basel-Stadt, a canton of Switzerland, covers the city of Basel and a few surrounding suburban municipalities (37 km^2 with a population of 200,000).

The administrative information system is concerned with:

- data about residents and enterprises
- financial data of the canton and the municipalities (taxes, wages for employees, etc.)
- data describing land and buildings (ownership, use, etc.)

This is a large application of the administrative type which has built up since 1970, using IBM hardware and the data base management system (IMS). It gradually became operational and is now successfully used in most departments of the administration. It can be used for scheduled processing of large amounts of data (batch mode). The interactive query and update operations at numerous terminals in many offices of the administration are important for the permanent updating of the data. It is now used for the preparation of reports and statistics for planning.

Geo-data processing system. From the very beginning, it was clear that the geometrical properties of objects should be included in the system, but it was not deemed practical to do this within the central computer system designed mainly for administrative tasks.

Therefore, an independent system, specially suited for the manipulation of geometric data, was built.

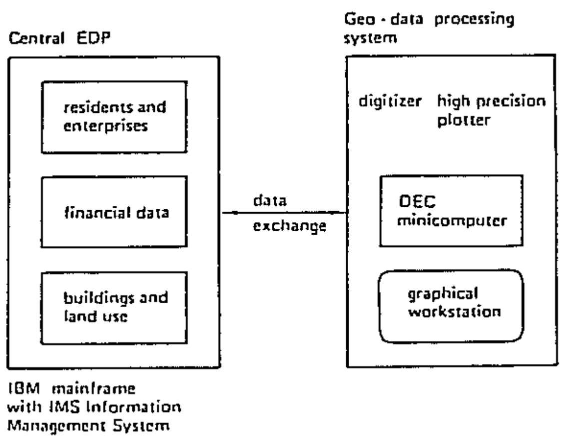


Figure 15:

This system is connected to the central data processing unit (Figure 15). The Clumis software (Clumis) has been built by the hardware vendor for this system. Basically, two different types of data sets are currently maintained:

- for the property map in scales from 1:50 up to 1:2500.
- for the town map featuring streets and street names for use by the general public (scales from 1:2500 up to 1:75000).

There are plans for inclusion of the existing utility line map (featuring water, sewage, gas, electricity and telephone lines) in the same scale as the property map.

Processing is not limited to the geometry oriented approach. The object concept is partially included as well. As an example: The area of a parcel may be computed using the stored geometry and transmitted to the central administrative data base. The parcel is recognized as an object in both systems, using the same identifiers. Other objects are streets, utility lines, etc.

Most of the attributes of the objects, whose geometry is stored in the specialized system, are stored in the central administrative data base. The geo-data system supports the following basic geometric and graphical units:

- points with coordinate values based on the national grid
- straight lines and arcs of circles
- map symbols
- text

For each geometric object a minimal number of attributes (main identification key and object classification) are included.

Access is through identifiers, location or interactive pointing on the graphical work station. To speed up spatial access, all data is divided into map sheets. The moment data within a sheet should be accessed, all data of the sheet is brought from the external compressed form to the internal form which is especially suited for spatial queries.

To maintain the data integrity across sheet boundaries, special tests are periodically performed.

The separation into two computer systems excludes a generally available interactive graphical query facility. The predominant use of the graphical system is the maintenance of maps.

Reasons for success. The most important points making this endeavor successful are only remotely related to the topics of this paper:

- a comprehensive plan for realization of EDP applications within the administration, based on data base principles.
- minimal redundancy
- a piece of data is stored only once and can be used for different applications.
- data security
- data protection (including the geometric aspects of reality as well)

A clear organization for the different functions of data collection and processing, especially concerning the operations for property surveying and measuring of utility lines is needed; the surveying department performs measurement operations in the terrain and maintains the data base.

In retrospect, it seems that during the development, sound theoretical concepts have been used. It could be concluded that the surveying practitioner, with his long experience of using geometric data, has had, all the time and partially unconsciously, the relevant concepts for describing the functions of a geo-data system.

Practical Advantages of the Methodology

A GDMS of the type proposed in the secton on efficient implementation would have brought the following advantages to the application described above:

1. The data base approach makes it possible to integrate geometric data with any other data. At the present, geometry is often confined to a separate graphical system limiting interconnections.

2. Data base administration would become possible with the data base administrator having conceptually suitable tools to enforce data quality.

3. With an unambiguous, complete geometry and procedures for easily evaluating implicit relationships, data base structure may be simplified. Maintenance is more automatic because less explicit relationships need to be updated.

4. A generally accepted conceptual framework would make the development of application systems easier. At the present time, each organization defines its own versions of the geometric concepts.

5. A good query language would make it possible for end-users to program simple applications themselves, thus increasing data usage.

6. Programming application systems would become much cheaper.

7. Because program development would be easier, the data collection and checking programs would be made more automatic with large potential savings and better data quality.

FURTHER RESEARCH

Despite the universality of geo-data management, its problems are surprisingly difficult and much research is needed to reach a consensus on a minimal set of concepts necessary for a wide-spread application of the methodology.

The following seem the most relevant problems to study:

1. The connection of the abstract data type and data base approaches into one data definition language

2. Definition of a complete set of basic geometric structures and formalization of the related integrity constraints

3. Formalization of the concept of geographical data independence

4. Definition of a general geo-query language

5. Efficient access to geo-data

Of course, the most pressing task is to test the concepts of the present article by applying them to as many Land Information Systems as possible. We advocate the constructive approach to this field of research. The proposed tools should be implemented as an operational GDMS and utilized in various applications. We believe that a realistic data description language can be developed only after this effort.

Actually, we proposed that the system structure of Figure 10 be built and tested "botton-up":

1. A DBMS should be used as a basis (Layer 5).

2. Next, a minimal set of structures and access functions for objects composed of nodes and lines should be implemented together with a spatial directory, locality of access and spatial queries (Layer 4).

3. Based on Layer 4, basic geometric structures may first be implemented as ordinary application programs. After gaining sufficient experience they can be formalized into a data description language as part of the GDMS (Layer 3).

4. The handling of application dependent objects is developed similarly to Layer 3 but might never attain the same level of formalization (Layer 2).

CONCLUSIONS

1. Both the object and geometry oriented views of geo-data are essential, and geo-data poses special requirements on data management. Completeness of data contents is the most important requirement for application design.

2. Figures 6 and 7 form a natural data model for conceptually defining geo-data contents and for indicating the various roles of geometry-related attributes.

3. Geographical data independence is a central requirement of a GDMS, be it tailor-made or built upon a DBMS.

4. Geometric integrity assurance is a main design objective for a geo-data management system.

5. The GDMS should be specified with a clear layer structure defined by the access functions of each level. One layer should correspond to basic geometrical structures.

6. A general query language for interactive data selection and representation is needed.

7. Geographical data types are natural to use and are needed as primitives in a GDMS. Separate points, lines and areas do not suffice as the only data types. More complicated aggregates, together with appropriate access functions, should be freely definable as basic geometric structures.

8. Spatial directories and locality of access are necessary for efficient performance.

ACKNOWLEDGEMENTS

The Finnish part of this work was funded by the Finnish Academy, the Data Centre of Hensinki Metropolitan Area and the Finnish Real Estate Register Project and was performed at the Laboratory of Information Processing Science of Helsinki University of Technology.

The authors wish to collectively thank their numerous colleagues in the Federation Internationale des Geometres, the Land Information Institute and the Spatially

Oriented Referencing System Association. Discussions and comments on the background material have greatly affected this paper.

REFERENCES

AS Australian Standard 2482 - 1981, Interchange of Feature Coded Digital Mapping Data, Standards Association of Australia, 1981.

Brüggemann, "Die 'Einheitliche graphische Schnittstelle (EGS)' im Vorhaben Automatisierung der Liegenschaftskarte", Nachrichten aus dem Karten- und Vermessungswesen Reihe, 1979, p. 7..

Burton, W., "Logical and Physical Data Types in Geographical Information Systems", Geo-Processing 1 (1979), North Holland Publishing Co., p. 167.

Clumis, Cadastral and Land Use Mapping and Information System, Ferranti Celtec Graphics Ltd., Edinburgh, Scotland.

Cox, N.J., B.K. Aldred and D.W. Rhind, "A Relational Data Base System and a Proposal for a Geographical Data Type", Geo-Processing 1 (1980), North Holland Publishing Co., p. 217.

Erstling, R. and M. Linder, BABEK: Grobkonzept für eine ALK-Schnittstelle, Report 5525/2061-1, Datum e.V. Bonn, Germany, 1979.

FIG, Federation Internationale des Geometres, Commission 3, Proceedings of the XVI Conference, Montreux, Switzerland, 1981.

Frank, A., "Application of DBMS to Land Information Systems", 7th International Conference on Very Large Data Bases, Cannes, France, September 1981, p. 448(a).

Frank, A., "Landinformationssysteme - theoretische und praktische Probleme", Paper 305.1, FIG XVI Congress, 1981 (b).

Frank, A., "MAPQUERY, Data Base Query Language for Retrieval of Geometric Data and Its Graphical Representation", ACM SIGGRAPH 1982 Proceedings, Boston, Massachusetts, 1982.

IBM, Geographisches Informationssystem, Anwendungs- und Systemsbeschreibung, Form GE 12-1506, IBM, Germany, 1979.

ICOREM, s.a., Cartographie Numerique pour la Gestion Automatisee des Donnees Foncieres Urbains; Realisations Francaises de Traitement Informatique des Donnees Cadastrales, Marseille, France, 1978.

ISOK, A System for Transferring Map Information Between Data Banks (in Swedish), Kommunförbundet, Stockholm, Sweden, 1980.

Kevany, M., "The User of Interactive Graphic Systems in Local Government in North America", Urban Data Management Symposium, 1979, the Hague, Netherlands.

Messmer, W., "Informationssystem, Datenbank und Mehrzweckkataster", Mehrzeckkataster, Mitteilung Nr. 21, Institut für Geodäsie und Photogrammetrie, ETH, Zurich, Switzerland, 1977.

SEMA Informatique, Silver-F Presentation/Specifications: Fonctionelles du System, Montrouge, 1979.

Tamminen, M., Management of Spatially Referenced Data; Conceptual Study of System Requirements and Structure, Helsinki University of Technology, Helsinki, Finnland, 1981.

Tamminen, M., "The Extendable Cell Method for Closest Point Problems", BIT, Vol. 22, No. 1. 1982, pp. 27-41.

Tamminen, M., "Efficient Spatial Access to a Data Base", Scholnick, N. (Ed.), Proceedings ACM SIGMOD Conference, Orlando, Florida, 1982.

UDMS, "Urban Data Management Symposium:, June 1981, Proceedings, Oslo, Norway, 1981.

Ullman, J.D., Principles of Database Systems, Pitman, London, United Kingdom, 1980.

USGS, U.S. Geological Survey, Computer Files and Attribute Codes for Digital Line Graphs, Washington, D.C., 1980.

Weber, W., "Three Types of Map Data Structures, Their ANDs and NOTs, and a Possible OR", Proceedings First International Advanced Study Symposium Topological Data Structures for Geographic Information System, Harvard, Massachusetts, 1978,