B F-7

Chapter 2

DESIGN OF CARTOGRAPHIC DATABASES

Andrew U. Frank

National Center for Geographic Information and Analysis (NCGIA) and
Department of Surveying Engineering
University of Maine
Orono, ME 04469 USA

1. INTRODUCTION

Computers were found to be very useful tools for helping with the production of maps. The contents of a map sheet can be stored in computer memory and changes easily introduced. An automatic plotter can then be used to redraw the map. Today, many organizations that have to maintain collections of maps from the traditional mapping agencies to public utilities, use automated mapping procedures in one form or another. The organization of storage of cartographic data in the most effective manner becomes then a topic that needs attention (Buchmann, et al. 1990), and criteria for optimization are necessary.

Cartographic data is voluminous and therefore storage should be compact. Compact storage not only saves money as less storage capacity needs to be bought but it also reduces processing time. This argument is less important today, as the reduction in cost for storage devices, primarily for magnetic disk devices, in the last years has been dramatic. In 1990, hard disk devices cost around US$15 per megabyte[3] and prices are falling further. Disk capacity is around 1 Gigabyte. The new optical disc technologies offer very inexpensive readers-prices are less than US$1000 and should fall further in the next few years. The medium is inexpensive, the production of a 600 Megabyte disk (similar in size to the Compact Discs used for music) is in the US$10 range and a number of distributors offer cartographic data in this format. It is apparent that storage space and cost

Frank, Andrew U. "Design of Cartographic Databases." In *Advances in Cartograpy*, edited by J.C. Muller, 15-45: Elsevier, 1991.

of the necessary devices is not a dominant concern any more and we need only worry about orders of magnitudes. It has been repeatedly observed that the cost of storage is dropping faster than most agencies can collect the data; i.e. the cost to buy all the storage devices necessary to store today's data is equal to the cost of the devices necessary a few years ago when the data was stored and will most likely be the same cost to store the vastly increased amounts of data in the future - storage cost per organization seems constant and independent of the increase in volume due to reduction cost by advances.

The storage size is also relevant for processing, as many operations require, in first approximation, time proportional to the amount of data to process (Samet 1989, p. 37). Fortunately, advances in processing speed have been very fast in the last few years, currently doubling in a single year, making this consideration less important. An exception is the delay to access data stored on disk, which has remained relatively constant during the past decade and will remain slow. Access to data on a disk takes about 20 to 50 milliseconds, that is about 10^6 times slower than access to data stored in main memory.

In conclusion, technical reasons do not limit the size of databases anymore. Data compression - transparent to the user - remains an interesting research topic, but saving storage space should not be a primary driving force in the design of a cartographic database.

On the other hand, we observe that collecting and maintaining the data is a major cost. Data collection is the primary investment and thus it is imperative that the data can be used effectively and for a long time. Our attention must therefore concentrate on methods to simplify and expand the use of the data and technology must be used to facilitate this goal. Experience over the last decade shows that technology advances are very fast. Hardware progress especially is generally faster than expected. Software advances are much slower, and to make matters worse, most people overestimate what can be done and how fast it will be ready for use - the so called 'software crisis' (Traub 1989). The least progress is made in understanding the organizational framework in which a cartographic database is constructed and used. It was shown that introducing new technology in an existing organization is difficult and slow. The technical solution should be to adapted to the organizational requirements; we should not try to force the organization into a technical solution that does not fit. Management of data in organizations is likely to be the challenge of the future (Egenhofer and Frank 1990, Frank 1990, Haywood, chapter 4).

In consequence, this chapter will concentrate on the logical issues that are likely to affect cartographic databases and how they are used within organizations for a number of years to come. It seems important to avoid spending time and effort on issues that will be resolved automatically by more powerful hardware, and to concentrate on the topics that will persist and need solutions based on a cartographers understanding of the goals, uses and limitations of cartography.

We first attempt to clarify the notion of a cartographic database and comparing it with the related term geographic information system. We then discuss different concepts for modelling a cartographic database, specifically one can understand a map as an image, the result of drawing instructions, a structured set of symbols or as a representation of map features. In the forth section, we then arrive at similar conclusions, considering the stages of the mapping process and the intermediate results of each. In the next sections, we investigate the datastructure and database requirements to link a GIS and a cartographic database and then the simpler case to construct multi-scale cartographic databases that can deal with map series. We conlude with two sets of recommendations, one for practicioners and one with open research questions.

2. WHAT IS A CARTOGRAPHIC DATABASE?

In the last few years a number of cartographic institutions have built databases or otherwise organized the collection of cartographic data they use to produce maps. Such collections are probably properly called cartographic databases. It must then be asked how they are different from geographic information systems, a term currently in wide used (Frank 1980). In this section, we try to give reasonable definitions for these terms. This is difficult in a field that is very young and in progressing rapidly and at a time that seems to follow Humpty Dumpty's rule 'When I use a word, it means just what I choose it to mean - neither more nor less' (Carroll 1960, p. 269), often without regard for etymology, taxonomic clarity and the established use of terminology in related fields.

2.1. Databases and database management systems

In very general terms, a database is a collection of data together with the methods to manage these data. Just a collection of data without the organization and methods for access etc. is not justly called a database. A database consists of the software needed for storing and retrieving the data as well as the data which have been stored (Figure 1) (Codd 1982, Date 1986, Zehnder 1981).

##Figure 1: Database concept (Frank 1988)##

The programming of the database software is necessarily very complex in order to respond to all the demands of storing and retrieving data in a safe and secure manner. The computer industry has produced Database Management Systems (DBMS) which are general purpose and fulfill generic database requirements. They are adapted by the database manager to a specific use, i.e. to manage a specific data collection.

A database management system should provide the following functions (Frank 1988):

- —Storage and retrieval of data; selection of data using different fields.
- —Standardizing access to data and separating the data storage and retrieval functions from the programs using the data. This makes database and application programs independent of each other, so that changes in the one do not necessarily lead to changes in the other.
- —A standardized interface between database and application programs isolate the application program form the details of the physical storage structure.
- —Allowing for access to the data by several users at a time.
- —Providing for the definition of consistency constraints for the data which will then be automatically enforced. Consistency constraints are rules which must hold for all data stored, and are an excellent technique to reduce the number of errors in a large data collection.

For a slightly different, more elaborate list see (Codd 1982).

##Figure 2: Three Schema Concept (ANSI/X3/SPARC 1975)###

The conceptual schema (Figure 2) describes the data and their relations at a logical level. It explains the potential linkages between the data and shows the potential for application programs and query languages to use the data. The physical or implementation schema maps the data which are described in the conceptual view to the storage devices. Separating logical from physical design allows the organization to maintain the conceptual structure of the data for a long time and adapt rapidly their mapping to accommodate changing hardware.

The conceptual schema adopts a 'corporate' view and attempts to show the logical integration of data from different organizational units and one can use multiple 'user schemata' to describe a conceptual view to each group of users that is more appropriate for their usage of the data (Frank 1990). Typically a user view does not contain all the data available, but only the parts relevant for the task. In addition, it may also rename or restructure data to allow people to use the data in a structure with which they are familiar. The database management system then automatically converts the data to this users 'view'.

The data must be considered a 'corporate' resource, and the more widely they are used the better. It is a general observation that economical benefits and thus a cost advantage of a database system are only attained if multiple user groups can use the same data _ the more users share in the fixed cost of data collection and data update, the less costs are incurred and the more beneficial the results. It must be possible for many people in an organization to access the same data at the same time and it may be even necessary for different groups to have well defined permissions in order to update the data concurrently. We also observe, that organizations acquire geographic data from other agencies (Frank 1990). This may lead to technical difficulties which can be overcome using concepts of transaction management (Date 1986, Ullman 1988).

It is not necessary for an organization to embrace a concept of centralization in order to reap the benefits of the database idea. Databases can be built conceptually centralized but physically decentralized, allowing each organizational unit to keep the data it needs but aiding or organized sharing of data (Frank 1990).

A database management system must include methods to guarantee long term usability of the data in order to protect the original investment of the data collection effort. This includes the logical consistency constraints, that guard against certain types of erroneous data. Methods must be set up to ensure that data, once collected, cannot be lost, despite malfunctioning of the hardware, disk failures or loss of electric current. The database management system must also help the organization to define responsibility for data and ensure that only authorized users update data elements following proper rules. They typically include controls to prevent unauthorized access to data. Cartographic data may be less sensitive and some mapping agencies less concerned with these problems, but a good database management system provides all these services - for very low additional cost - and they can occasionally be very useful.

2.2. Databases and information systems

Both terms 'database' and 'information system' are often used in similar context - is there a difference? Before we access this issue, it is helpful to look first at the difference between data and information.

By **data** we understand sets of symbols in a form that is accessible to computer-based data processing. Computers can use these data and process them according to programs. The data can completely change its form, appearance, size but it continues to remain data. **Information** on the other hand, is 'an answer to a human's question', a piece of data that at this moment is meaningful to a person. The human's mind interprets the data in relation to the world and draws some conclusions, decides on action or similar. It is always a good idea to recall that computers do not make decisions! As an aside, we propose to use the word **'document'** for a collection of data in a format not currently accessible to a computer - so a map collection, records from an exploration, etc., are 'documents'. They need human interpretation to be meaningful, whereas the data are formalized such that programs can process them.

A database is thus a collection of data with its management software. The data in the database is accessible from other programs, but not directly by end users. The clients of the database are the programmers and the application programs they write. The information system delivers information to users, and it must therefore include methods (programs) for the user to communicate her 'information needs' and then other methods to present the data that make up the answer in a format that is understandable for end-users. Thus an information system is more encompassing than just the database (see Fig.1) and includes specialized applications, query languages (Egenhofer and Frank 1988, Frank 1982) and cartographic output routines.

2.3. Spatial Databases

Both GIS and cartographic databases deal with the management of **spatial data**, which we define as data that has a relation to a spatial location (Frank 1987)

Spatial data = Geometry + Attributes

Cartographic database and GIS deal with spatial data similarly to systems for CAD/CAM, VLSI etc.(Brodie 1984, Plouffe, et al. 1984, Stonebraker and Guttman 1984, Udagawa and Misoguchi 1984, Wilkins and Wiederhold 1984). It has been observed, that managing spatial data poses some special problems and that DBMS designed for the management of administrative data are not well suited to manage these non-standard data (Härder and Reuter 1985). We first address the functionality necessary for spatial data handling, specifically location-based retrieval. Afterward we discuss how we model geometry to represent spatial data adequately.

It has been found that these non-standard applications ask for a larger set of functions than the standard commercial applications. There are a similarities among them in their requirements, but it is not clear if a single comprehensive DBMS for all these applications could or should be built. The cost of fulfilling the specific demands of spatial applications are high and an application that does not use one or the other function should not have to bear its cost in terms of performance, hardware requirements, etc. Also, implementations need to be designed for the specific load, e.g. VLSI data has different characteristics than cartography data. It was proposed that a non-standard database should be built from a common kernel and extensions that can be freely combined to meet the demands of a specific application (Schek and Waterfeld 1986). Recent discussion has concentrated on the applicability of the object-oriented concept to databases (Dittrich 1988, Dittrich and Dayal 1986), to spatial databases and the specific problems of spatial access methods (Buchmann, et al. 1990).

Efficient retrieval of spatial data based on spatial location is a special problem and some specific solution have been proposed. The problem is generally stated as

##"Retrieve data for all objects within <rectangle>"##

where the rectangle (Figure 3) is given by the coordinates of two points.

##Figure 3: Query rectangle Figure 4: A minimal bounding rectangle##

In order for such queries to be answerable, the system must contain the minimal bounding rectangle (Figure 4) for each object (Frank 1981) or another method to express position and extension of an object in a generalized fashion, e.g. bounding circle.

In order to be able to process such a query, which is technically speaking a two dimensional range query, special data structures are necessary and a large number of methods have been developed (Barrera and Frank 1989, Frank 1983, Frank and Barrera 1990). For an overview see (Samet 1989) and for a recent conference on the subject consult (Buchmann, et al. 1990).

In systems that do not use such data structure, the database is divided into map sheets or facets and the user is aware of these. Special processes must be provided to join sheets and continue the objects from one to the other (Beard and Chrisman 1988). A **seamless database** not divided into sheets is necessary for a GIS and certainly attractive for a cartographic database (Chrisman 1990).

It is difficult to assess the differences in performance of these spatial data structures, as they depend on specific properties of the data loaded and their performance seems to be relatively similar (Kriegel, et al. 1990). The problem not yet properly addressed is to integrate these spatial data structures with the other parts of the comprehensive spatial DBMS.

The database discussion of the past few years has concentrated on the relational data model (Codd 1982, Ullman 1982) and many implementations are available today. Many commercial GISs allow one to store attribute data in relational databases and link them logically to the geometric or graphic data. The capability to connect to existing data sets stored elsewhere is important, but makes preserving consistency during update operations more difficult. It has been proposed to store all data, geometric and attribute, in a relational database (Abel 1988, Waugh and Healey 1987). But so far no convincing demonstration including a performance report on a realistically large data collection has been published. One major problem that remains is the large number of slow disk accesses that add up to unacceptable delays, when retrieving the large data sets typically necessary to display a map.

It also appears difficult to fit the complex data structures of spatial reality into the restricted relational model (Roessel 1987). The relational data model was found to be quite restrictive, structuring all data in tables (relations) of a single type of tuples (records) with a fixed set of attribute values. Spatial data can be structured in this form, even if this is not necessarily the most natural description for spatial concepts and geometric data models (Frank 1990, Goodchild 1990)(Figure 5).

##Figure. 5 Relational data structure to implement the topological data model (Herring 1990)##

The relational algebra on which relational databases are founded, process a relation at a time. This does not accord with the record at a time processing of the currently used programming languages. Programmers of spatial data handing routines, especially for geometric and topological operations can not often exploit the power of relational processing and must resort to slow single record retrieval and processing. The difference in concepts between database and programming language further complicates the transfer of data values between the two.

The object-oriented paradigm used in software engineering and programming language design (Meyer 1988) appears to be applicable to databases and promises to provide a more flexible method for structuring complex data (Egenhofer and Frank 1987). It is, similar to relational algebra, soundly based on a mathematical formalism, a property the earlier hierarchical and network data models missed (CODASYL 1971). An object- oriented data model should allow generalization hierarchies (probably several ones at a time, so called 'multiple inheritance') and methods to construct complex objects from parts (Atkinson, et al. 1989, Stonebraker, et al. 1990). This appears to correct one of the major problems with the relational data model's breaking conceptual objects in minute pieces (Nievergelt 1990) and makes the object-oriented data model most promising for spatial databases. Unfortunately, commercial quality object-oriented DBMS are just emerging and a number of research issues are still open (Kim 1990).

2.4. Cartographic databases and GIS

A simple definition for a cartographic database is a database which contains cartographic data together with the management software necessary for its collection, update and output.

This begs the question what is **cartographic data**, a question which will further lead into the essence of designing cartographic database. To begin with, it is sufficient to state that cartographic data is data that when rendered represents maps and their graphics in one or another format. From cartographic data, a map can be produced or reproduced easily. A cartographic database is thus a database system - as defined above - that contains cartographic data, i.e. map data, and the procedures to display maps, either on a screen or on paper. These maps can then be used for different purposes.

In contrast, a **geographic information system** is an information system which can respond to a wider array of questions regarding some aspect of geography, of information about land, its use, ownership or population. The response can be in the form of a map, but can also be a report, a table or a textual answer.[4]

The major difference between a cartographic database and GIS is in what is modeled. A GIS contains a 'model of reality'; this model is limited to some specific tasks and the major data needed for them. The cartographic database contains models of maps (which, in turn, are models of reality)(Chrisman 1983, p. 303, Frank 1984).It must be admitted that in practice there are no sharp distinctions and many systems which are advertised today as GIS are in effect more like cartographic databases and some of the cartographic databases existing do allow some extra flexibility in their use.

3. What is stored in a cartographic database

In order to discuss the design of a cartographic database we have to understand what kinds of data should be stored. Database design should always proceed from a conceptual understanding of the data and from some rough assessment of the size of the data collection, to a discussion of the data structures to be used and then finally to the physical organization of the data.

What is stored in a database is described in the database schema. The database schema language provided with one or the other generic DBMS is more or less capable of describing the data to be stored accurately and naturally (i.e., without any unnecessary artifacts). It is then that the descriptive powers of the data model and the database schema language become apparent.

A map can be viewed from different levels of sophistication, and depending on the viewpoint, different elements and structures become apparent (Chrisman 1983, p. 303-304). On a first level of differentiation, we can see a map as an image, consisting of pixels, as a line drawing, or as a complex communication structure.

3.1. Maps stored as images

A simple view of cartographic data is considering maps as images. The map surface is divided into regular cells (pixels, short for 'picture elements') and for each of them the color and intensity is recorded. Such data can easily be collected by scanning existing maps and the maps later reproduced from it. This is extensively used in navigation and military applications. The major drawback is the immense amount of data to be stored and processed even if compression methods are used. For example, a map sheet of 40 x 60 cm, scanned with a minimal resolution of 1/10 mm - about FAX quality - and 4 levels of intensity produces 6 Megabytes of raster data per color and this at a very modest quality; better quality resolution, e.g.. 1200 dpi[3] ≅ 1/50 mm, and 256 levels of intensity results in 1.5 Gigabyte[4]. This is probably not the most desirable method to store the thousands of large scale maps covering a large country.

More limiting than the immense amount of data to be handled are the limits on the operations which can be performed on such data: all the standard image processing operations, from averaging to edge enhancing are available - but there is only limited use for them in cartography. Images are well suited for 'electronic editing' at the pixel level, and this is used in high quality commercial map production work. Hand drawn manuscripts and other maps are scanned and then electronically combined and edited. If images from maps of different scales are stored, even a limited type of zoom is possible. Systems have been proposed that would support command and control operations and would allow 'zoom' and 'pan' over a seemingly seamless map (Tuori and Moon 1984). If the resolution is reduced to what is reproducible on a PC graphics screen (pixel ca. 1/10 mm), and some efforts in data compression etc. made, then a number of maps can be stored on a CD-ROM and with appropriate software they can be retrieved and used for a backdrop behind additional information managed in a separate format.

On a raster image, we cannot automatically find locations by name or conditions on its location relative to other features. The system can be augmented with a gazetteer that lists named locations and their coordinates, and a translation algorithm from geographic coordinates to image coordinates. Then map displays can be retrieved by location name, but this is not an operation applied to the raster image properly, and uses additional data sets [Neal, 1989]. On raster images, it is difficult to follow a line through intersections with other lines, and therefore all tasks of route planning etc. require human interaction to find the continuation of a line after a complex road intersection. It can also be difficult to find the area of certain complex figures when it is necessary to find the continuous boundary line. In general, raster images limit the analytical capabilities enormously. On could use the theory of symbolic projection (Chang 1990, Chang, et al. 1990) to cartographic image data and create indexes in form of strings. This would require to segment the images into clusters of pixels that form one object - a notoriously difficult task in cartography but would then avail additional retrieval logic to the user (Chang, et al. 1988).

3.2. Vectorized data

Maps can be seen as a set of lines and points (disregarding color filled areal features) and stored as such. Such vector data can be produced automatically from scanned maps by extracting the vectors from the raster. The necessary skeletonization process that proceeds vectorization proper has a tendency to deform junctions (Figure 6).

##Figure 6: Skeletonization of a T-junction##

Traditional rules for map symbolization do not always facilitate automatic digitalization. Separation of lines based on color and then assigning them a feature code is one of the few methods for automatic encoding of features. Unfortunately, this is hindered in cases where contour lines are printed in brown in general, but blue for glaciers and lakes and black in rocky areas (e.g. some European topographic maps). Further, contour lines are often interrupted, not only to place index height figures, but also for buildings, etc. Filling these gaps automatically is difficult if not impossible.

At the concept level, the map is still an image to be reproduced by vectors and vectorization is essentially a form of data compression but does not change applicable operations compared to image data. Names and labels are just sequences of vectors and not yet letters or symbols for which one can search.

3.3. Graphics command files

A map is drawn on a computer as a sequence of commands, which direct the plotting equipment to draw lines, characters etc. A map can also be stored in this form, including commands to draw vectors in different line styles, text and symbols. Such a format is quite compact and at least for storage, quite efficient. It is the 'obvious' choice for maps originally produced with an automated cartography system. As early work was primarily carried out at institutions that produced maps, this was the typical solution for a on data exchange standard had been influenced by this. The data structures are simple linear sequences of coordinates, with some header records indicating the feature class (Canadian Council on Surveying and Mapping 1982, Zarzycki 1984). In principle this is still quite satisfactory for a system that primarily maintains maps, if one is aware of its limitations.

Data for such collections is also produced by digitizing existing maps using special programs which allow editing and cleaning of the data. The human digitizing operator interprets the drawing (the map document) to create a data set that has some higher level structure. For example symbols and labels are encoded. Maps stored in this format can be updated with relative ease. The appearance of a symbol or the type face used can be globally changed - which is not possible if we store the map as an image. The existing systems are primarily oriented toward map maintenance and do not provide extensive query facilities. It is not clear what would be the potential for access to the database with an ad hoc query language.

3.4. Maps as structures

A map is not just an image, at least not to a cartographically literate person: there are clearly distinct features which are considered separately within the spatial context. Most map users clearly see the advantage of using a road map over a remotely sensed image for the planning of a driving route - a road map contains the relevant facts (and few others) encoded in a format that is useful for route planning. In a remotely sensed image, the relevant information must first be extracted from a much larger, not yet filtered, collection.

[TAB04[[TAB04]One approach for structuring cartographic data is to consider cartography as a language with its own syntax and vocabulary (Dacey 1970, Robinson and Petchenik 1976). The formalization of this concept to a degree that is useful for cartographic data processing is slow (Palmer and Frank 1988, Youngmann 1977). It is clear that there is some regularity in the transformation of information content into graphical symbols (Bertin 1983, Imhof 1972, Mackinlay 1986), which enables the map reader to correctly interpret the meaning of the symbols. However, we have not yet found a comprehensive set of rules that help to select these transformations (Buttenfield and McMaster 1991). If we consider the map as a collection of symbols which communicate some meaning and not as an image, we must be able to group the graphic elements into the meaningful symbols and describe the relations between them - a very difficult problem (Head 1990, Schlichtmann 1985). The database schema then contains these symbols and their structure and not the graphical primitives with which they are drawn.

At this level, we may separate the map proper, the cartographic symbols that describe the world, and the context, that assigns meaning to the symbols and makes them understandable (Youngmann 1977,). A structured description of the 'reference' part of a map is needed and some initial work was done in (Ehrich, et al. 1988, Neumann 1988).

The 'information' part of the map can in a first attempt be structured in point features, line features, and areal features (Morrison 1988, p. 24). For each feature there is an indication of its location, either in map or world coordinates, assuming that the transformation between the two is defined by a mathematical projection. This location is most often an approximation to the location of the real world entity on the surface of the earth, but it need not be. Maps also depict graphically, and thus locate spatially symbols for things that do not have a geometric location in reality, from labels for highways to country names, not to mention the symbols on thematic maps. Features are encoded using elaborate classification schemas, mostly based on cartographic (topographic) traditions (Canadian Council on Surveying and Mapping 1982), but other methods would be possible (Guttenberg 1981). There is sometimes a labeling text (e.g. the spot height, the name of the town) with a position for writing this text.

The U.S. draft standard for digital cartographic data (Morrison 1988) lists geometrical structures for cartographic objects, giving terms like line segment, string, arc, etc. for linear features and area, polygon etc. for areal features. These constructs should be sufficient to express the content of a map drawing, such that data can be exchanged from one system to another and be reproduced. They may also be a good base to start the design of the data structures of a cartographic database.

Unfortunately a single feature is not always mapped by a single map symbol nor does a graphical map symbol represent only one feature. One of the major problems of cartographic data structures is how to deal with the linkages between individual graphical map objects and the features. The structure of such a file approaches a state where it becomes a model of reality (i.e. a GIS): the coordinates are world coordinates, the feature classes are real object classes. There is a very subtle but crucial difference in understanding what one represents and for what purposes it exists. If one wants only to represent a map in a digital form, such that it can be edited, communicated and reproduced with computer systems, one is at one extreme. If one wants to construct a general purpose database from which to produce maps of varying content and scale and for different purposes one is in the middle and the database for general purpose spatial analysis, is at the other extreme. Again, one has to respect the special quality of a cartographic database and 'one cannot expect a database which was generalized using procedures for cartographic generalization to be a reliable source of predictable quality for analysis' (Brassel and Weibel 1988) and one could add, but also not render a database prepared for analytical tasks and expect a quality map. The acid test is often, to determine, if the database contains purely graphical data which has no counterpart in reality (e.g. the location of labels).

4. Mapping process

In order to understand what is represented in a cartographic database, it may be useful to consider briefly the cartographic process. A cartographer goes through a number of steps to construct a new map and at each one applies different types of rules to different types of objects. These steps are roughly (Rase, chapter 9):

—conceptualization of a map topic and content,
—collection of data: data from different sources, most often maps, are assembled.
—design of the map: the cartographer decides which part of her mental model of reality is to be shown on the new map and selects the topics to be rendered
—design of the map style: the cartographer decides on the map symbols, color schemata etc.
—selection of the individual map features
—rendering of map features

She/he optimizes for various goals at the same time mostly to assure maximum communication. This includes most aspects of the map generalization process. The cartographer does often not follow a strict linear sequence despite the fact that theoretical studies generally follow a linear sequence model. In principle one could deduce automatically the cartographic database from the GIS, similarly as the manual cartographer establishes his data set from various sources (Figure 7).

##Figure 7: Deduce map database from GIS##

Unfortunately, none of the steps of these processes are well understood and formally described. Despite some efforts, only very limited parts are automated. A few topics have found attention, especially line generalization (Beard 1988, Douglas and Peucker 1973), label placements for point features, (Freeman and Ahn 1984) and generalization of built up areas [stauffenbiel- thesis]. One of the major problems is the lack of objective measure for the qualities of the map one optimizes for and which are affected by the generalization process (João 1991). For example we lack a measure for the 'density' of a map, but do also not have an operational defined 'optimal density' an programmed process could use. We also do not have appropriate methods to describe the 'scale' of a cartographic data set. There is agreement that the publication scale is not describing the essence, but it seems so far the only usable method.

As a gedanken experiment, let us assume for a moment that we have a fully programmed cartographic process, such that we can automatically produce maps from a GIS. It is not clear if such a fully automated process can ever be achieved, but there have also not yet been criteria set, what would be an acceptable level of performance. The problem appears to be 'AI complete', meaning that it requires 'understanding' (Brassel and Weibel 1988, p. 231) and if it could be solved, many other problems of artificial intelligence, e.g. understanding of natural language text, could be solved. However, it appears entirely feasible, that partial solutions can be found that solve a majority of the cases and produce results that are usable even if they are not of as high a quality as the best examples of current manual cartography. These processes could be made available to human cartographers and used under their control to support their work.

Continuing with the gedankenexperiment, if the cartographic process was fully automatic, we need only store a GIS as complete and as true to reality as possible from which we could deduce any map desired with no manual effort. There would, in principle, not be a need for a cartographic database. If we do not have such an automated map production process, there is a need to store the intermediate results, because they are the product of human effort and should it should not be wasted. We assume here that human labor is a valuable good, that needs to be economized and preserved, whereas computer processing power is essentially free and programmed steps can be repeatedly done. Thus we see that a cartographic database can have any of the following contents:

—Cartographic model of the world, defining the real world entities which are of importance to map making (as opposed to e.g. real estate taxation). This essentially defines a map topic, a selection of object types to be included but this data can still be used for analytical processes.
—Collection of data describing real world entities which should be mapped under certain circumstances, e.g. at a certain scale. This is the selection of individual features to be shown.
—Symbolized cartographic features, with labels positioned etc., including many of the results of map generalization processes.
—Drawing instructions to produce a map.
—Image of the map.

First, one notices the similarity of this list with the results of the previous section. From this list it appears, that the ultimate goal of a cartographic database is probably the 'cartographic model of the world' but the lack of a fully automated mapping process forces us to store intermediate steps. It seems as if the more on comes closer to the final product, the less flexibility there is. This is largely correct, but the process is, as mentioned before, not linear. Decisions in each step have substantive effects on the total appearance of the map: for example a simple change in the cut of the map sheet requires extensive adjustment of objects that were previously affected by the map boundary and the objects that are now affected as the accommodation for graphical problems at the map border must be undone and redone (Figure 8).

##Figure 8: A change in map sheet size is affecting name placement##

The cartographic process is concerned with the management of one piece of information cannot be used for another one. Thus the spatial interaction of cartographic symbols on the map influences what must be selected and where the appropriate symbol must be placed. The human cartographer tries to optimize this for maximum communication with the map user. Recording these elements of the graphical image makes the data collection less versatile and less useful to produce other maps from the same data. Efforts underway to capture these rules (Buttenfield and McMaster 1991) may eventually help to overcome these limitations.

Many of the current efforts are directed not towards a fully automated mapping process, starting from a geographic information system or the generic cartographic database, but towards the slightly different goal, namely the deduction of one map from an already existing map data set, typically involving a change in scale (Beard 1987). This is certainly easier, if the given map is similar in scale theme, etc. to the desired on, but not quite feasible today either (Dobson 1988).

5. Linking a GIS and a cartographic database

Unless we can produce maps from a GIS without manual intervention by a cartographer and need maintain only one database (namely the GIS), we have to live with multiple databases that contain data about the same objects. The critical problem is not the original preparation of the derived data sets, but long-term maintenance. Without costly and error-prone maintenance the data sets are out of date quickly and thus useless. For the following discussion, we will assume that the GIS is updated regularly in order to fulfill its analytical or administrative purposes. We will frame the problem in terms of automatic synchronization of the cartographic databases with the GIS updates. A change in the GIS database should be checked and, if necessary, the cartographic database notified automatically updated.

The cartographic database contains internal representations for specific maps that need to be produced and the GIS contains models of the same objects which can be used for other ends, for example spatial analysis. These two databases need not, but can be separated and stored in different places. Nevertheless we can consider them as a logical unit and study their data models and schema. The methods for their synchronization are not substantially more difficult if the two databases are stored on different computers, the so called 'distributed database' situation (Frank 1985, Mohan 1984).

In database jargon, one could consider the carto-database as a ünther 1989), i.e. a specific way to present the contents of the database (thus in the database jargon a 'view') which is not produced on the fly but is materialized (i.e. stored). This is a common method to avoid repeated costly processing - or in this case, processes which are not fully automated and to save the human effort that went into construction of the view. The problem with materialized views is that they need to be updated with any change in the database (Blakeley, et al. 1986, Hudson 1990).

The cartographic database contain all the data necessary to produce the map - just as any other cartographic database. It also contains linkage data that connects the cartographic features with the corresponding objects in the GIS (Broom 1989). These linkages are complex and need research attention, as they are not always just simple links between a GIS object like a building and the corresponding map feature. A forest feature on a map may be composed from a number of forest parcels, or an airport building appearing as a single feature in the cartographic database but as several objects in the GIS.

In its simplest form the linkage has no other objective than notification of change. Any change in the GIS is then translated into a warning flag associated with the relevant features in the cartographic database. If the cartographic database is used to produce a new version of the map the cartographer's attention is drawn to the flagged features and he individually and manually updates the cartographic features, which are then updated and stored in the cartographic database. The cartographer should have access to the updated data in the GIS, but there is no specific requirement or need for sophisticated generalization software; if it exist, it will be increase the cartographers productivity.

This notification of change should occur independently of the reason or the program used to change the GIS - any change should result in this notification. The functionality must thus be included in the DBMS software and not in the application programs of the GIS. In the database community one calls this a 'trigger', which guards some database object and if a predetermined condition occurs (e.g.. the object is changed) the trigger fires and a set of operations is executed, e.g. the cartographic database is notified of the change. The database research community has investigated systems to achieve such effects in databases; result from the research in 'active databases' (Chakravarthy 1989) should be used for the cartographic database problem. The database researchers are usually concentrating on triggers in a single database - not loosely coupled databases like a GIS and a cartographic database - and have studied specially time-critical (less than one second) propagation of values (Dayal, et al. 1988) -

again not of concern in the cartographic database situation, where a notification within weeks is most likely sufficient. The requirements for the GIS and cartographic databases are thus slightly different. There is an immediate need to explore the exact demands and formalize specifications in order for the database research community to include our needs in their work (Smith and Frank 1990).

Practically, one will observe that many objects in the GIS and the corresponding features in the cartographic database contain the exact same data. This depends on the model built in the GIS and the cartographic design. The more similarity between the purposes of the GIS and the map, the more correspondence between the data. The correspondence may become so large, that one forgets about the functional difference between GIS and cartographic database. This may be dangerous as there are most likely cartographic features which must be adapted for graphical representation; one is then tempted to tweak the GIS data to produce the correct cartography and lose the correct model of reality in the process.

A better solution would be to define a procedure that copies the data for the GIS in all cases where no change is necessary and uses data that is separately stored in the cartographic database in all other cases. The GIS data then serves as a 'default' value which can be overridden by previously determined and stored cartographic data. Such a method can be extended to include some filtering of the data transformation or other programmed processing and only if this is not sufficient, specific values need to be stored and maintained. Current commercial DBMS do, however, not contain methods in their data models which facilitate this solution and extensions should be added. One can assume that such facilities would be useful for other applications than just cartography and should therefore be constructed as a general purpose solution.

6. Cartographic databases for map series

Many of the map producing agencies and firms which are interested in a cartographic database produce not only a single map - why worry about a database then? - but produce multiple maps organized into series, which cover the same area and similar themes, e.g. for different uses or at different scales. Examples include the U.S. Geological Survey that produces several topographic map series with different scales, but as well the private cartographic companies which maintain data sets for road maps of different scale, map sizes etc. The same cartographic features are then rendered in multiple maps in slightly different forms. It is a current challenge to develop database schemata that can represent this situation.In this case we deal with maintaining map data only, not linking map data with GIS data. A cartographic data model is used throughout and differences between the map series are limited. More feature data appears the same on the different maps and considerable savings can be achieved by not copying the data. But again, not the savings in storage space are the most important, but the reduction in expenses to update the map database is the major benefit.

No actual implementation of such a multi-series cartographic database has been reported. Most organizations that maintain multiple map series are either not yet using computer aided systems and maintain independent graphical masters for each map in each serie, or are just starting to use computer aided cartography. Even the few that have used computer cartography tools for a while maintain separate data files for each map sheet in each serie, exactly paralleling the traditional organization (Dobson 1988). This is a common first step in automating any manual process, but should be overcome in a second phase, when one goes to study the special capabilities of a new technology and tries to see how they can be best taken advantage of. In this case it is the possibility to maintain multiple data set with complex linkages between them in a coordinated manner which allows independent exploitation of each data set individually but permits transmittal of changes from one to the other. Modern DBMS provide such facilities, they should be used to help solve the difficult problem of maintaining map series.

As a practical example, one of the U.S. commercial mapping companies maintains a large collection of road maps, covering North America with many series of different scale, sheet cut, additional information etc. They boast to include 19,536 changes since the previous year's edition in one single product (McNally 1990). Changes are mostly addition and deletion of road segments, changes in road numbers etc. which for a large part, affect multiple map sheets and must be manually applied to each - not to forget the administrative effort to assure consistent updates of all maps. Replication of data in this example is not only between the different map series but also within one single serie, where each map sheet shows in this product a single state with some parts of the surrounding states; thus a change in a position close to a state boarder may affect one or more maps of adjoining states.

Most often there is a difference in scale between map series. It is often said, that digital cartographic data does not have a scale. Technically this is correct, as from one set of coordinate values (even if they are already scaled) a map in any other scale can be drawn - transformation of coordinate values from one map scale to another are simple. But as little as photographical reduction or enlargement can change map scale in a wide range, can computed coordinate transformations. Each map or data file representing a map has an intrinsic 'map scale' which was used to determine map content, symbolization, level of generalization (Beard 1988). Currently we do not understand very well these limitations and cannot characterize a map data file other than by the scale for which it was constructed - a situation similar to the 'lineage' information that is used to characterize other data quality aspects of a data file, to which it is related (Chrisman 1985, Morrison 1988, p. 132).

We need to design a data structure for such a multi-series cartographic database and our first focus is on a data structure which supports multiple levels of detail. Computer scientist have studied a number of data structures that allow to quickly retrieve the few salient aspects but also allow retrieval of increasing amounts of detail. Such structures are often build using the 'divide and conquer' approach. Examples are quadtrees and strip trees (Samet 1989, Samet 1989). In such data structures data is represented such that a rough approximation is retrieved first and, with additional effort, is gradually improved. In a quadtree areal data is represented with an increasingly finer subdivision of squares. Rough approximations to an area can be constructed by only considering a few top levels of the tree and not access the detailed levels. This not only reduces detail in the perimeter but also excludes all areas smaller than a threshold. Combined with a method to smooth the boundaries results that seem to be quite acceptable for certain applications may be achieved (Bjørke and Aasgaard 1990).

The less well known strip trees (Ballard 1981) is a based on similar concept, but applied to line segments. A line is repeatedly approximated by two line segments, connecting the end points with the point further away from the straight connection between the end points. This can be applied recursively to the two newly constructed line segments. A line generalized to a certain level is retrieved by following the tree of approximations down to a certain level. This results in a line generalization somewhat similar to the well known algorithm described in (Douglas and Peucker 1973). It seems easy to design a similar data structure to structure a collection of point symbols such that an increasing number are selected the deeper one searches.

These assume that

—the more refined representation is the a superset of the less refined one, and
—one uses a regular method for subdivision and refinement.

Both these assumption are not applicable for multi-scale cartographic data. In cartography, a map of larger scale is not just the data from the smaller scale map plus additional detail. We have to account for the cases where a number of features of one kind are grouped together to form a generalized feature (e.g. a number of building symbols are replaced by a single town symbol) - the generalized feature replacing the collection of details. Each feature has an associated map location where it is to be shown. Even for features that show on more than one map, their location may vary slightly from one to the next scale - to accommodate other graphical map elements and to avoid overcrowding. The data structure would have to use the values from the smaller scale map unless there are specific instructions stored with the larger scale level, each transition marked with the scale where the change or refinement occurs. For example on may see a sequence as in the following example (Figure 9).

##Figure 9: Changes in the data from scale to scale##

In maintaining a multi-scale cartographic database one can not identify a unique entry point for changes. Not all changes are entered in the database at the lowest scale and then percolate up. Changes are entered in the map sheet that is next to be produced and it seems desirable that no effort is spent at that time to update the other sheets potentially affected - the cartographer needs to concentrate on the map he is working on to achieve a consistent product. Thus one should automatically detect what other maps could be affected and post a warning flag. If these sheets are to be re-edited, the cartographer sees all the updates that were posted on related sheets and he incorporates them as necessary in a manual editing step.

The methods sketched above appear to be practically useful, could be realized within a short period of time and promise considerable benefits. They do not require any methods of automatic cartographic generalization to work. Not only the reduction in storage space that could be achieved counts, but more the simplified maintenance procedures. Assuring that thousands of small changes are consistently applied on all maps is a major organizational problem.

7. Practitioner's summary

A cartographic database is a collection of data used for preparing maps together with the software to manage this data. Such databases are increasingly established by the map producing agencies and private companies and the data sets together with the appropriate software will be made available commercially to others that want to produce maps. Such databases will be organized in terms of computer graphics primitives (text, polyline etc) and be ready to reproduce maps of acceptable quality.

A cartographic database is different from a GIS as a GIS is constructed for a multitude of applications and supports analytical procedures as well as some mapping; the GIS is intended to model a subset of reality geared to a specific set of tasks. The cartographic database is designed to manage map data.

A cartographic database can contain raster images of maps, produced by automated scanning of paper maps, it can contain vectorized map images, map drawing instructions or feature data. One arrives at similar differences if one considers the cartographic process and the intermediate products. Depending on the conceptual model used, different operations are available to the user. A difference easy to observe is the availability (or absence of) a query language. There the conceptual data structure is reflected in the kind of queries that can be executed. All users must always bear in mind the purpose a data collection was established for and its limitations. Using a cartographic database for analytical work is likely to produce misleading results and using a database established for analytical purposes to produce maps will result in a graphically inferior map.

Database management system are the computer science tools to organize complex collection of data. They should be used to organize the storage of map data in a systematic fashion and help to protect the data for long time usage. The major investment is in the data collected, not in the hardware or software that manges it. Graphical map data have been used and reused for long periods of time and to achieve the same in the electronic age, we have to concentrate on the conceptual data structures in order to guarantee that the data collections can survive several changes in hard- and software. The database management tools help to achieve this.

One can logically link a GIS and a cartographic database for a region and establish connections between the objects in the GIS and the features in the cartographic database. If the GIS is updated the related features in the cartographic database can be flagged and cartographers are warned about changes and will then manually effectuate the necessary cartographic changes. In such an arrangement one may also reduce the duplication of data between GIS and cartographic database and use the GIS for mapping purposes. This requires a 'default logic' and data structures which allow to store specific map data that overrides or complements the GIS data, for example if features must be moved on the map to avoid graphically congested areas. The cartographic database, not the GIS, will also include positions for labels etc.

Somewhat similar organizations could be use to manage the cartographic data for multiple map series of differing scale, content etc. covering the same area. Current such cartographic data sets are mostly stored independently in a file per map sheet. A database could add links between cartographic features representing the same objects in different maps and facilitate maintenance and guarantee the consistency of updates. From such a scheme gradually methods of computer assisted generalizations could be used to reduce the data to be stored and reduce manual efforts. This seems immediately feasible, does not require any scientific breakthroughs and promises considerable benefits.

8. Projective summary

A database must respect the inherent structure of the application area it deals with. Thus the most important aspect of a cartographic database is to understand its structure based on understanding the cartographic production and communication process. We are severely lacking in this. Currently cartographic databases are mostly established in terms of computer graphics output primitives. A more appropriate structure could start with understanding maps as a language and using a cartographic grammar as a means for means of structuring the database (Brassel and Weibel 1988, p. 235). The structure revealed by a grammar is in principle translatable into a database structure.

Despite a copious literature on map generalization (for an overview see (Beard 1988) and (Brassel and Weibel 1988)) very little is known well enough that appropriate computer programs can be written. Quantitative descriptions of map data and quantitative characteristics of different types of map data or cartographic processes are mostly lacking. An notable exception is Buttenfield's effort to characterize cartographic line data in order to guide the appropriate line generalization algorithm and its parameters (Buttenfield 1989). Except for precision of point and contour lines, we currently lack methods to describe quantitatively the quality of cartographic data etc. and abuse 'scale' and expectations of the current mapping practice to characterize cartographic data sets. Progress in methods to describe map data are urgently needed and will benefit generalization studies, as we gain quantitative goals for the optimizations involved.

Another line of investigation is to analyze the architecture of a cartographic production software and identify what steps are necessary in the map rendering process, how they are structured and what data is necessary for each. Using principles of software engineering to organize processing in layers with similar functions and achieve minimal interaction between modules will reveal the structure of the data needed and how they are interrelated.

An immediate line of research and development should be to structure the multi-representation cartographic database that are necessary. This is not only a matter of practical concern but yet another approach to analyze cartographic data structure, this time starting from real cartographic data and its analysis. The goal would be to design a data structure that can manage the data sets necessary to produce the map sheets of several map series, which cover the same area, but at different scale or with differences in map styles. Two types of benefits can be reaped, namely the reduction of data storage, due to reduction in redundancy and second, more important, simplifications in the map update process.

Acknowledgements

Funding from NSF for the NCGIA under grant SES 88-10917 is gratefully acknowledged.

References

Abel D (1988) Relational Data Management Facilities for Spatial Information Systems. In: Marble D (ed.) *Third International Symposium on Spatial Data Handling*. Sydney, Australia, :

ANSI/X3/SPARC (1975) Study Group on Database Management Systems, Interim Report 75-02-08. *SIGMOD FDT* 7 (2)

Atkinson M, Bancilhon F, DeWitt D, Dittrich K, Maier D and Zdonik S (1989) *The Object-oriented Database System Manifesto*. ALTAIR Technical Report No. 30 - 89. GIP, ALTAR, Lechesnay, France.

Ballard D H (1981) Strip Trees: A Hierarchical Representation for Curves. *Communications of the ACM* 24 (5 (May)) : 310-321.

Barrera R and Frank A U (1989) *Analysis and Comparison of the Performance of the Fieldtree*. Report Orono, ME: Department of Surveying Engineering, University of Maine.

Beard K (1987) How To Survive A Single Detailed Database. In: Chrisman N R (ed.) *AUTO-CARTO 8, Eighth International Symposium on Computer- Assisted Cartography* 1987. Baltimore, MD,

Beard K (1988) *Multiple representations from a detailed database: a scheme for automated generalization*, University of Wisconsin - Madison.

Beard M K and Chrisman N R (1988) Zipper: a localized approach to edgematching. *The American Cartographer* 15 : 163.

Bertin J (1983) *Semiology of Graphics*, Madison, WI: The University of Wisconsin Press.

Bjørke J T and Aasgaard R (1990) Cartographic zoom. In: Brassel K and Kishimoto H (ed.) *4th International Symposium on Spatial Data Handling* July 23 - 27 1990. Zurich, IGU: 859-868.

Blakeley J A, Larson P-A and Tompa F W (1986) Efficiently updating materialized views. In: Zaniolo C (ed.) *SIGMOD '86* may 28- 30 1986.

Washington DC, ACM SIGMOD Record: 61-71.
Brassel K E and Weibel R (1988) A review and conceptual framework of automated map generalization. *International Journal of Geographic Information Systems* 2 (3) : 229-244.
Brodie M L (1984) CAD/CAM Database Management. *IEEE Database Engineering* 7 (2) : 12-20.
Broom F (1989) Multiple representations - U. S. Bureau of the Census Position. In: Buttenfield B P and DeLotto J S (ed.) *Multiple Representations: Initiative 3 Specialist Meeting Report.* Santa Barbara, CA, NCGIA: 72-73.
Buchmann A, Günther O, Smith T and Wang Y-F (1990) *Design and Implementation of Large Spatial Databases - Proceedings of the First Symposium SSD '89, Santa Barbara CA, July 1989,,* New York: Springer Verlag.
Buttenfield B and McMaster (1991) *Rule based cartographic generalization,* in preparation.
Buttenfield B P (1989) Scale-dependence and self-similarity of cartographic lines. *Cartographica* 26 (1) : 79-100.
Canadian Council on Surveying and Mapping (1982) *National Standards for the Exchange of Digital Topographic Data - Draft Report,* Ottowa: Energy, Mines and Resources Canada.
Canadian Council on Surveying and Mapping (1982) *National Standards for the Exchange of Digital Topographic Data - Draft Report,* Ottowa: Energy, Mines and Resources Canada.
Carroll L (1960) *The Annotated Alice. Alice's Adventure in Wonderland and Through the Looking Glass,* Hardmandsworth, England: Penguin Books Ltd.
Chakravarthy S (1989) Rule Management and Evaluation: An Active DBMS Perspective. 18 (3) : 20 - 28.
Chang S-K (1990) Visual reasoning for information retrieval from very large databases. *Journal of Visual Languages and Computing* 1 (2) : 41-58.
Chang S-K, Jungert E and Li Y (1990) The Design of Pictorial Databases Based Upon the Theory of Symbolic Projection. In: Buchmann A, Günther O, Smith T R and Wang Y-F (ed.) *Design and Implementation of Large Spatial Databases.* New York NY, Springer Verlag: 303 - 324.
Chang S K, Yan C W, Arndt T and Dimitroff D (1988) An Intelligent Im-

age Database System. *IEEE Transactions on Software Engineering* (May) : 681 - 688.

Chrisman N (1985) An interim proposed standard for digital cartographic data quality: supporting documentation. In: Moellering H (ed.) *Digital Cartographic Data Standards: An Interim Proposed Standard.* Columbus OH, National Committee for Digtial Cartographic Data Standards:

Chrisman N R (1983) The role of quality information in the long-term functioning of a geographic information system. In: Wellar B S (ed.) *Auto-Carto 6* oct. 16-21 1983. Ottawa, 303-312.

Chrisman N R (1990) Deficiencies of sheets and tiles: building sheetless databases. *International Journal of Geographical Information Systems* 4 (2) : 157-167.

CODASYL (1971) *Data Base Task Group Report,*

Codd E F (1982) Relational Data Base: A Practical Foundation For Productivity. 25 (2) : 109-117.

Dacey M F (1970) Linguistic aspects of maps and geographic information. *Ontario Geographer* 5 : 71-80.

Date C J (1986) *An Introduction to Database Systems*, Reading, MA: Addison-Wesley Publishing Company.

Dayal U, Blaustein B, Buchmann A, Chakravarthy U, Hsu M, Ledin R, McCarthy D, Rosenthal A, Serin S, Caery M J, Livny M and Jauhavi R (1988) The HiPAC Project: Combining Active Databases and Timing Constrains. *SIGMOD RECORD* 17 (1) : 51 - 70.

Dittrich K R (1988) *Advances in Object-Oriented Database Systems--- 2nd International Workshop on Object-Oriented Database Systems,* Bad Münster am Stein-Ebernburg, FRG: Springer Verlag; Berlin; New York.

Dittrich K R and Dayal U (1986) *Proceedings International Workshop in Object-Oriented Database Systems,* Washington, D.C.: IEEE Computer Society Press.

Dobson M W (1988) Digital Cartography in the World of Commercial Publishing. In: Marble D (ed.) *Third International Symposium on Spatial Data Handling* 1988. Sydney, Australia, International Geographical Union: 1-8.

Douglas D H and Peucker T K (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer* 10 (2) : 112 - 122.

Egenhofer M J and Frank A U (1987) Object-Oriented Databases: Database Requirements for GIS. In: Aangeenbrug R T and Schiffman Y M (ed.) *International Geographic Information Systems (IGIS) Symposium: The Research Agenda* Nov. 15-18 1987. Crystal City, VA, NASA: 189 - 212.

Egenhofer M J and Frank A U (1988) Towards a Spatial Query Language: User Interface Considerations. In: DeWitt D and Bancilhon F (ed.) *14th International Conference on Very Large Data Bases* Aug. 1988. Los Angeles, CA, 124-133.

Egenhofer M J and Frank A U (1990) Prospective Views of GIS Technologies and Applications. In: *Symposium Brasileiro De Geoprocessamento* 1990. São Paulo, Brazil, Escola Politécnica da Universidade de São Paulo: 95 - 102.

Ehrich H, Lohmann F, Neumann K and Ramm I (1988) A Data Base language for Scientific Map Data. *Geologisches Jahrbuch* A (104) : 139- 152.

Frank A U (1980) Landinformationssysteme - Ein Versuch zu einer Abgrenzung (Land Information Systems - An Attempt of a Demarcation). *Nachrichten aus dem Karten- und Vermessungswesen.* Frankfurt a.M., F.R.G., Institut für angewandte Geodäsie:

Frank A U (1981) Applications of DBMS to Land Information Systems. In: Zaniolo C and Delobel C (ed.) *Seventh International Conference on Very Large Data Bases VLDB* Sept. 1981. Cannes, France, 448-453.

Frank A U (1982) MAPQUERY: Database Query Language for Retrieval of Geometric Data and its Graphical Representation. *ACM SIG Computer Graphics* 16 (3) : 199 - 207.

Frank A U (1983) *Probleme der Realisierung von Landinformationssystemen; 2. Teil: (Problems in the realization of Land Information Systems; Second part:) Storage Methods for Space Related data: The FIELD TREE [reprint of the paper in M. Barr (ed.) Spatial Algorithms for Processing Land Data Symposium 1983].* Bericht 71. Zurich, Switzerland: Swiss Federal Institute of Technology, Institute for Geodesy and Photogrammetry.

Frank A U (1984) Computer assisted cartography: graphics or geometry? *ASCE Journal of Surveying Engineering* 110 (2) : 159.

Frank A U (1985) Distributed Databases for Surveying. *Journal of Surveying Engineering, American Society of Civil Engineers* 111 (1) : 79-88.

Frank A U (1987) Overlay Processing in Spatial Information Systems. In: Chrisman N R (ed.) *AUTO-CARTO 8, Eighth International Symposium on Computer-Assisted Cartography* March 29-April 3 1987. Baltimore, MD, 16-31.

Frank A U (1988) Requirements for a Database Management System for a GIS. *Photogrammetric Engineering & Remote Sensing* 54 (11) : 1557- 1564.

Frank A U (1990) *Acquiring a digital base map - A theoretical investigation into the problems of sharing data.* Report (in preparation) Orono, ME: Surveying Engineering Department, University of Maine.

Frank A U (1990) Spatial Concepts, Geometric Data Models and Data Structures. In: *GIS Design Models and Functionality* Mar 21- 22 1990. Leicester, UK, Midlands Regional Research Laboratory, University of Leicester:

Frank A U (1990) Telecommunication and GIS - Opportunities and Challenges. In: Cavill M (ed.) *Networking Spatial Information Systems* May 30 - 31 1990. Melbourne, Telecom Australia: AF-1 - AF-9.

Frank A U and Barrera R (1990) The Fieldtree: A Data Structure for Geographic Information Systems. In: Buchmann A, Günther O, Smith T R and Wang Y-F (ed.) *Symposium on the Design and Implementation of Large Spatial Databases.* New York, NY, Springer-Verlag: 29-44.

Freeman H and Ahn J (1984) Autonap - An Expert System for Automatic Map Name Placement. In: *International Symposium on Spatial Data Handling* 1984. Zurich, Switzerland, 544-569.

Goodchild M F (1990) A geographical perspective on spatial data models. In: Maguire D (ed.) *GIS Design Models and Functionality* 21 - 22 March 1990. Leicester, Midlands Regional Research Laboratory: