Cognitive and Linguistic Aspects of Geographic Space (90-13)

Egenhofer, Max J.
Kuhn, Werner
McGranaghan, Matthew

1990

Cognitive and Linguistic Aspects of Geographic Space

Preprints of three papers by:

Max J. Egenhofer,
Werner Kuhn, Andrew U. Frank
and Matthew McGranaghan

National Center for Geographic Information & Analysis
and
Department of Surveying Engineering
University of Maine
Orono, ME 04469

National Center for Geographic Information & Analysis/NCGIA

Report 90-13

Table of Contents

Preface-and -Acknowledgements

The three papers contained in this report address different aspects of the problem of formalizing human communication about geographic space. Much of this thought grew out of the National Center for Geographic Information and Analysis's Research Initiative #2,"Languages of Spatial Relations". Each of these papers were presented at a NATO Advanced Study Institute on Cognitive and Linguistic Aspects of Geographic Space, held in Las Navas del Marques, Spain, during July 1990. The meeting was directed by David Mark and coordinated by Andrew Frank. These papers will appear in a volume resulting from the meeting. They are being published as an NCGIA technical report to make them accessible before they are published and printed in that volume.

The first paper, entitled "Deficiencies of SQL as a GIS Query Language", argues that SQL and various extended versions of it are not adequate geographic query languages. They lack the integration of graphical display in retrieval and presentation of query results and do not support the set operations necessary for spatial query. In the second paper, "A Formalization of Metaphors and Image-Schemas in User Interfaces", an algebraic approach to formalization of interface metaphors is presented as a step toward the design of metaphor-based interfaces. This approach to mapping source to target domains is demonstrated by analyzing the metaphorical and image-schematic bases the "zoom" function. The third paper, "Matching Representations of Geographic Locations", sets out fundamental differences in the representations of geographic space and spatial relations in minds, written location descriptions, and conventional cartographic data sets, and proposes a geographic data structure which might facilitate establishing correspondence between locations in each representation. Together, these papers range across theoretical and practical concerns in incorporating “spatial sense" in GIS.

This report presents work that has been supported by a grants from the National Science Foundation (SES-88-10917), Digital Equipment Corporation (grant No. TP765536), Intergraph Corporation, and a NATO Advanced Study Institute Grant The support of these organizations is gratefully acknowledged.

DEFICIENCIES OF SQL AS A GIS QUERY LANGUAGE[1]

MAX J. EGENHOFER
National Center for Geographic Information and Analysis
and
Department of Surveying Engineering University of Maine
Orono, ME 04469, U.S.A.
MAX@MECAN1.bitnet

ABSTRACT. Numerous proposals have been made to extend the relational database query language SQL to serve as a spatial query language and currently efforts are under way to establish a standardized spatial SQL. Here it is argued that the SQL framework is inappropriate for an interactivespatial query language and an extended spatial SQL is at best a short-term solution. The serious deficiencies of any spatial SQL are divided into two groups:

- the severe difficulties to incorporate necessary spatial concepts into SQL such as graphical display and its specification and

- the lack of power within the relational framework with missing support for complex objects, object identity, meta queries, and non-first-order logic --- all crucial when dealing with spatial data.

1. Introduction

The Structured Query Language SQL is enjoying much popularity in the database world and has become the standard for relational database management systems (ANSI 1986). Although originally designed as a high-level interface to relational database management systems (Chamberlin et al. 1976) to manipulate relational tables (Codd 1970), SQL is sometimes even considered as *intergalactic dataspeak* (Stonebraker et al. 1990) for many other applications. Frequently, the underlying power of the relational calculus has proven to be insufficient and numerous extended SQLs have been proposed such as SQL dialects for dealing with time and date (Datel988), temporal (Ariav 1986) and historical (Sarda 1990) data, and dynamically defined complex objects (Lorie and Schek 1988).

The application of SQL for spatial data handling has been discussed as well. The lack of a spatial concept in the design of SQL makes any use of *vanilla* SQL for spatial data handling simulate spatial concepts in terms of the few, predefined non-spatial concepts (Westlake and Kleinschmidt 1990). Currently, there is little consensus among researchers about the application of SQL for spatial data as demonstrated by the number of different SQL extensions. The SQL dialects tailored to spatial applications (Roussopoulos and Leifker 1985, Sikeler 1985, Ingram and Phillips 1987, Egenhofer 1988, Herring et al. 1988, Roussopoulos et al. 1988, Ooi et al. 1989) differ considerably in (1) the extent they cover spatial properties, (2) the degree at which they formally define the semantics of the extensions, (3) their syntactical implementations, and (4) the degree at which they comply with the standardized SQL structure. At the same time, they all struggle with incorporating spatial concepts into a framework designed for data modeled as 2-dimensional tables.

This paper argues that these problems are due to SQL's inappropriateness as a framework for a high-level spatial query language. It is not discussed whether or not it is possible to express some spatial queries in SQL. Definitely, SQL can be used for querying any data modeled in a tabular format, and spatial data can be mapped onto such tables (Waugh and Healey 1987, Abel 1988); however, such a representation is not the most natural form for modeling spatial data and leads to unnecessarily complex queries. Comparing database query languages with programming languages, such a low-level treatment of spatial data is comparable to breaking a RECORD type in a high-level programming language into physical address assignments in Assembler language.

The scope of this paper is to underline this position with a number of arguments demonstrating severe shortcomings in the idea of extending the SQL framework to the query language for spatial data. Our findings are based upon extensive studies of spatial query languages carried out by researchers at the University of Maine in requirements analysis (Frank 1984, Pullar 1987, Egenhofer and Frank 1988a, 1988b), query language design (Frank 1982, 1984, Egenhofer 1991), and prototyping (Egenhofer 1984, 1989b, Egenhofer and Frank 1990). The focus will be exclusively on the role of SQL as a *query language* in its literate sense and any other roles, e.g., as a *manipulation language*, will not be discussed. To further restrict the focus, only one of the many interpretations of a query language will be considered. Currently, the use of SQL as a query language is overloaded and too many different tasks are performed with SQL. Some users consider it as a high-level interface language to tie a database management system into an application program; others use SQL as an interactive *ad hoc* query language; for another group of users, SQL is the data exchange

---

[1] This research was partially funded by grants from Digital Equipment Corporation under Sponsored Research Agreement No. 414 and TP-765536, and Intergraph Corporation. Additional support from NSF for the NCGIA under grant number SES 8810917 is gratefully acknowledged.

language to transfer data from one database to another. Unfortunately, very different requirements exist for each of these roles of a query language. For example, the interface between database and application program must fit will into a high-level programming language, while an *ad hoc* query language must consider the interaction between humans and computers. In order to avoid further confusions of different requirements and expectations, the discussions in this paper are limited to the use of SQL as an interactive spatial query language.

The remainder of this paper is structured as follows: first, a brief description of SQL is given. Section 3 focuses on the role of a spatial query language within a geographic information system (GIS), a particular range of applications in the realm of spatial data handling, and typical concepts GIS users employ when they interact with geographic data. Section 4 reviews how various spatial extensions have been implemented in SQL dialects and stresses the diversity of the approaches. The impediments of the SQL framework for a high-level, interactive spatial query language are analyzed in Section 5. The conclusions propose the design of spatial database interface languages, in lieu of query languages, to better integrate the interaction between user and database.

2. SQL

SQL (Chamberlin et al. 1976) is an implementation of the five relational operations selection, projection, Cartesian product, set union, and set difference (Codd 1970)---plus a few useful extensions for operations upon tuples such as aggregate functions. The syntax framework is the SELECT-FROM-WHERE clause which corresponds to the relational operations of projection, Cartesian product, and selection, respectively. Set union and set difference can be explicitly formulated among several SQL queries. For example, given two relations parcel and road with the respective attributes `parcel.number, parcel.owner, parcel. roadname`, `road.name`, and `road.width`, the following SQL query is to retrieve the owners of all parcels which are located on roads narrower than 15 feet:

```
SELECT       parcel.owner
FROM         parcel, road
WHERE        road.width < 15 and
             parcel.roadname = road.name;
```

The query result is a relation which is in an interactive environment by default presented in form of a table on the screen. More extensive discussions of SQL can be found in database textbooks (Date 1986, Korth and Silberschatz 1986, Ullman 1988) and SQL tutorials (Date 1989a, 1989b).

3. Spatial Queries

SQL has been successfully used as a query language for applications from the commercial realm which can be easily expressed in terms of tables, frequently called *standard database applications*. So-called *non-standard* applications (Harder and Reuter 1985), such as CAD/CAM, VLSI design, image databases, or geographic information systems, require a more complex data model than tables. A crucial criterion for evaluating the usability of a query language for a non-standard application domain is: "How useful are the database operations provided by the query language for the particular application?" (Schek 1988). The following list of typical queries in a GIS environment will help to assess the usability of SQL as a spatial query language:

- "Display a map of Las Navas."

- "What is the shortest path from Castillo Magalia to the train station?"

- "Where is the nearest bus station?"

- "In which direction is El Escorial?"

- "What is this?"---and the user points to some place on a screen displaying some graphical rendering.

Obviously, spatial queries refer to particular spatial concepts (Egenhofer and Frank 1988b) in a spatial data model (Frank 1991a), e.g., geometric objects with a complex internal structure, spatial relationships to select objects of interest, complex graphical display, and selection by pointing. GIS users base their spatial queries upon such concepts. Therefore, they are critical criteria in a spatial query language; however, neither SQL nor the relational data model support them. Relations lack the proper set of fundamental operations to properly support the majority of geometric and pictorial operations (Joseph and Cardenas 1988) and relational operations per se are insufficient to solve some typical GIS queries (Frank 1982, Egenhofer and Frank 1988b). For example, important problems, such as queries over combined maps, cannot be solved unless specific support of spatial operations is provided (Laurini and Milleret 1988) and spatial data are frequently selected upon their spatial location, rather than by references to some values.

4. Spatial SQL Dialects

Several proposals have been made to turn SQL into a spatial query language. The most significant extensions will be reviewed subsequently to provide the reader an overview of what has been accomplished, but also to show how the individual extensions proposed differ. This evaluation considers a spatial SQL dialect (Sikeler 1985), the pictorial query language PSQL (Roussopoulos and Leifker1985, Roussopoulos et al. 1988), Spatial SQL (Egenhofer 1987, 1988, 1991), GEOQL (Ooi et al. 1989), and the SQL-based GIS query languages for KGIS (Ingram and Phillips 1987) and TIGRIS (Herring et al. 1988),

4. 1. SPATIAL DATA TYPE

Almost every spatial SQL extends the domains of the relational calculus with *spatial data types*. Although there have been attempts to design a spatial query language exclusively based on the standard domains in relational calculus (Go et al. 1975, Berman and Stonebraker 1977), i.e., on integers, reals, and character strings, there is a general agreement that users of spatial query languages need a high-level abstraction of spatial data. An attribute over such a spatial data type will be referred to as a spatial attribute and a relation with a *spatial attribute* will be called a *spatial relation*.

What varies among the different spatial SQL dialects is (1) the spatial data model for which the data types are used and (2) the degree at which this extension is integrated into the SQL environment. The variety of spatial data types proposed for spatial SQLs includes

- a universal spatial data type (Sikeler 1985, Herring et al. 1988, Ooi et al. 1989);

- data types for each spatial dimension, i.e., points, lines, areas, and volumes, and their generalizations to a dimension - independent spatial superclass (Egenhofer 1988)--- essentially, the universal spatial data type above;

- a number of data types for a multitude of spatial properties such as area, perimeter, and length (Ingram and Phillips 1987) and the graphical representation as well; and

- a data type for bitmap graphics (Roussopoulos and Leifker 1985).

The actual syntax of these extensions differs considerably. Three particular approaches have been pursued:

- Some query languages reserve specific attribute names for spatial attributes such as loc (Roussopoulos and Leifker 1985, Sikeler 1985) or *map, area,* and *perimeter* (Ingram and Phillips 1987).

- Others assign no explicit spatial attribute to a spatial relation, but implicitly assume its existence (Herring et al. 1988, Ooi et al. 1989).

- Others define only the names of the spatial data types and with the data definition, spatial attributes are defined by associating an attribute name with a spatial data type (Egenhofer 1988).

4.2. SPATIAL RELATIONSHIPS

The extension with spatial data types is useless unless the pertinent operations and relationships are also defined. Spatial relationships are boolean operations to check whether or not a particular predicate holds true between tuples of two or more spatial relations. SQL tends to overload predicates, i.e., an operation may have multiple implementations and the system selects one depending on the type(s) of the argument(s) of the predicate. While such an overloading of SQL's standard predicates is sufficient for some applications, e.g., temporal relationships (Allen 1983), it is insufficient for spatial relationships. The set of standard predicates in SQL is too small to cover all spatial relationships (Egenhofer 1989a, Egenhofer and Herring 1990). For example, SQL lacks predicates for topological concepts, such as *neighbor* and *intersect;* therefore, syntax extensions are necessary to allow users to formulate queries with spatial predicates.

Similar to the extension with a spatial data type, all spatial SQL dialects include such spatial operations and spatial relations. Four major differences in the use of spatial operations and relationships can be observed:

- The number of spatial relationships covered by the individual SQL extensions varies considerably and different terminology is used as well.

- With two exceptions (Herring et al. 1988, Egenhofer 1991), no formal definitions are given for the semantics of the relationships.

- Spatial relations are introduced in prefix (Ingram and Phillips 1987) or infix form (Egenhofer 1988, Ooi et al. 1989).

- Spatial predicates are used upon spatial attributes (Egenhofer 1988) or upon relations (Sikeler 1985, Ingram and Phillips 1987, Herring et al. 1988, Ooi et al. 1989).

4.3. GRAPHICAL DISPLAY

An *ad hoc* spatial query language requires that query results can be displayed graphically. The graphical display of query results has two separate components in a query language: (1) specifying that the query result (or parts of it) should be graphically displayed --- as opposed to the implied tabular representation of alphanumeric data in SQL --- and (2) describing how to represent the query result. For both issues, different solutions have been proposed.

- A qualifier for a spatial attribute in the SELECT clause specifies that a particular relation be graphically represented (Sikeler 1985).

- All spatial attributes in the SELECT clause are displayed, while non-spatial parts of the query result are shown as alphanumeric tables (Ingram and Phillips 1987).

- The query result is displayed according to the status in a display environment (Egenhofer 1991).

The description of the graphical display of the query result has drawn only little attention. Most spatial query languages disregard this part or use only a default representation. Only PSQL and Spatial SQL propose solutions for this problem.

- PSQL displays query results according to predefined definitions, called picture lists (Roussopoulos and Leifker 1985) --- without providing a language to create or modify a picture list.

- Spatial SQL displays spatial query results according the definitions in the *graphical representation environment* (Egenhofer 1991). A comprehensive display language as a superset of SQL allows users to describe the use of colors, patterns, symbols, etc. for spatial relations.

4.4. SELECTION BY POINTING

Each SQL query is a stand-alone instruction without any reference to the previously asked queries or their results. Likewise, query results are always represented as a single "rendering" and no interaction with the currently displayed result is possible. For a spatial query language with graphical representation of query results, this SQL feature is a major restriction, because the graphical representation of query results animates users to refer to the drawings when formulating further queries.

Since SQL has no provisons for input other than typed characters, some spatial SQL dialects include an operator to identify a spatial object by pointing to its spatial location on the screen. Most commonly, pointing is implemented as a keyword, e.g., MOUSE (Ingram and Phillips 1987) or PICK (Egenhofer 1988). Such references may occur in WHERE clauses when a user refers to *this* object and uses a pointing device to select the object from the screen drawing.

4.5. COMPLIANCE WITH THE SYNTAX FRAMEWORK

Any SQL extension suffers from the dilemma of preserving the standardized form of SQL while extending SQL's functionality. A variety of syntax modifications of the SELECT-FROM-WHERE framework have been proposed in order to enable users to query spatial data. Three fundamentally different trends can be found:

- the addition of new clauses in which particular spatial properties are addressed, such as WITH LOCATION (Sikeler 1985) and AT (Roussopoulos and Leifker 1985) for spatial conditions, and ON (Roussopoulos et al. 1988) to specify an output format;

- the modification of the SQL framework by treating relations in lieu of attributes both in the SELECT and WHERE clauses and canceling the FROM clause which becomes superfluous; and

- minimal extensions within the given framework to comply with standard SQL (Egenhofer 1988, Ooi et al. 1989) and the definition of a *display environment* outside of SQL (Egenhofer 1991).

5. Impediments of the SQL Framework for Spatial Query Languages

Any SQL extension which attempts to comply with the SQL standard has to live with the conceptual and syntactical constraints set forth by the standardized framework. This section will show why it is so difficult to extend standard SQL so that it becomes a useful spatial query language. A number of shortcomings of the SQL framework for spatial data handling will be examined, some of which are due to the attempt of extending a given standardized query language. Two types of deficiencies are distinguished: (1) the lack of spatial concepts in the SQL framework and (2) the lack of expressive power to formulate types of (spatial) queries. All these shortcomings are crucial impediments for a spatial query language; however, it is not the lack of a particular functionality that makes the SQL framework inappropriate for a spatial query language. It is rather the fact that a great number of deficiencies exists for which no easy or simple remedies can be provided. Some of these problems may be fixed in SQL with considerable modifications of the initial SQL language, such as a recent proposal to integrate some syntax mechanisms for recursive query formulation into SQL (ANSI-ISO 1989). For other problems, no such "solutions" are in sight. Of course, such extensions are only patches and do not help to make SQL a sound and coherent spatial query language.

5. 1. DECOUPLED RETRIEVAL AND DISPLAY

Standard SQL is targeted almost entirely to retrieving data modeled in a tabular form. It stresses the functionality of formulating complex and powerful queries through the implementation of the relational algebra operations and, at the same time, disregards how to present the query result to the user. Only a single, default representation is provided--- the display in the form of a table. This uneven balance between retrieval and display is characterized by two features of SQL:

- The SELECT clause is overloaded with the projection of the attributes onto the resulting relation and the implied tabular representation of each query result. Such a combination may be appropriate for those applications which always require the results to be presented as tables only; however, it is a major impediment for a query language with renderings other than tables, e.g., drawings, or with a choice of renderings (Egenhofer 1987).

- The retrieval language is de-coupled from the representation of query results. SQL neither memorizes what has been displayed nor does it allow the users to formulate queries with respect to the query results displayed.

These shortcomings are most apparent when users request to display the query results in a non-tabular format.

5.1.1. Specification of Graphical Display.

The potentials for graphical variations of objects on drawings are much greater than those for manipulating the frame of a table, e.g., by adding headers over the columns or changing the sequence of columns. Graphical display involves the use of different colors, patterns, symbols, etc. (Bertin 1983).

What makes the *display specifications* more difficult is that a set of these graphical properties may be assigned not only to the entire result of a query, but also to specific spatial objects or classes of objects in the result. Especially for GIS applications with high-quality map output, these display specifications are too complex to be integrated into the actual query statement.

At a first glance, it may appear as a viable solution to separate a spatial query into two parts: (1) the instruction to retrieve the data wanted and (2) the *subsequent* command to display the query result. The query language --- or better the *retrieval language*--- would describe what to retrieve and a display language would specify *how* to graphically represent the query result previously retrieved; however, such a separation does not take into account that the query and specification of graphical representation frequently depend upon each other. For instance, given the following retrieval instruction:

```
SELECT          road.geometry
FROM            road, district
WHERE           district.name = "Avila" and
                road.geometry INSIDE district.geometry
```

After its processing, further information will be necessary in order to draw the major roads with a different line style than the remaining roads. In essence, more *queries* are necessary to separate the intermediate result --- the geometry of *all* roads---into two sets of roads so that the elements of each set can be represented with the same graphical attributes. Therefore, the demand for alternative displays is not solved by just drawing the spatial attributes in the SELECT clause. It must also be considered that the graphical display of query results may require more information from the database than what has been provided by the query result.

5.1.2. Combinations of Graphically Displayed Query Results.

Spatial query results may be depicted as drawings at which users look; however, a more dynamic interaction with query results is also necessary (Egenhofer and Frank 1988b). Users want to refer to the current drawing by asking further questions about it or they want to modify the current drawing, e.g., by adding further information to it or by removing information displayed (Egenhofer 1990). Such a dynamic interaction may be richer than the interaction with tables and requires support from the query language. The SQL framework supports only the retrieval of data based upon input typed by the user and the presentation of the data retrieved for the user, and no provisions have been made for an alternative retrieval, i.e., one partially based on the currently displayed result. Extensions of the standardized SQL framework to incorporate multiple representation types are essentially impossible, unless changes are made to a degree that the extension is no longer compatible with the standard.

5.2. POWER OF THE LANGUAGE

The power of the SQL language to retrieve data is determined by the operations of the relational algebra plus some additional concepts such as aggregate functions. Although SQL is *relationally complete* (Ullman 1982), i.e., any manipulation of tables is possible with a combination of the operations provided, it does not support some fundamental concepts, such as meta-data queries and higher-order queries. Furthermore, the retrieval part of SQL is inherently value-based, i.e., all operations are performed upon attributes and tuples can be only compared for equal values, but not for identity. These problems are general SQL problems which apply to non-spatial applications as well. While they may be less critical for traditional SQL applications, they impose serious restrictions upon the use of any spatial extension of SQL. The impact of these deficiencies on a spatial query language is discussed below.

5.2.1. Meta-Data Queries.

The SQL query facilities are designed for retrieving data, but they are limited for asking queries about the structure of particular tuples. *Data queries* typically start with some knowledge about the structure of the tables, i.e., users remember the names of the tables and the corresponding attributes. The instructions are to find corresponding tuples the values of which fulfill certain constraints. While these queries may cover the majority of questions users ask against a database, there are other kinds of queries with which users request information about data, e.g., "To which relation(s) does a particular tuple belong?", "To which attribute(s) does a specific value belong?", "What is the domain of a specific attribute?", and "To which view does a particular relation belong?"

These *meta-data queries* are particularly important for spatial databases when information is graphically represented about which users request further information. Topographic and thematic maps, for example, typically contain symbols and labels for various features. Users may point to them asking "What is this?" and expecting an answer like "This is a castle" or "That is a train station." The answers to such queries are the names of relations or attributes, not particular values of attributes.

Answers to meta-queries can be combined from data and meta-data. For instance, an answer to a query about a label on a map may be "This is the mileage along the main road from Las Navas to Avila." This answer contains meta-data --- the mileage is the attribute name of the label --- but also data, namely that the label identifies the mileage for a road and that it is the value from Las Navas to Avila.

Another geographic application shows the use of meta-data queries about the domain of an attribute. The query "What are possible soil classifications?" refers to the domain of an attribute *soil type*. Note that this query is different from "What are all the values stored in the attribute *soil type*?" The result of the first query is independent of the actual values stored in a table and its definition is the subject of the data definition, while the result of the latter query may change with the storage, deletion, or update of any value.

5.2.2. Knowledge Queries.

Queries of a nature similar to meta-queries are *knowledge queries* (Motro, and Yuan 1990), i.e., those used for explanations of the reasoning process of a query language. Spatial relationships, for example, are typically derived from a representation in a spatial data model, rather than being explicitly recorded (Davis 1986). The derivation is based upon rules which formalize the criteria for the individual relationships. Spatial information systems that allow users to tailor their spatial predicates (Herring et al. 1988) must also include facilities to let them inquire about the rules used for a specific predicate. For example, the adjacency between two parcels may be defined such that they share at least one common boundary, but have no common interior (Egenhofer and Herring 1990) and users may want to know from the system "Why did you identify the two objects as neighbors?" ---"Because they share a common boundary." Such knowledge queries about the characteristics of operations may also be extended to include their properties. For example, the query "When Avila can be reached by train from Las Navas, can Las Navas be reached from Avila by train as well?" is a query about the *symmetry* of the relationship *reachable---by --- train.*

In a similar way, users may ask queries about consistency constraints. For example, the query "What are the geometrical constraints about rivers?" may be answered by "They must not cross each other and a river cannot cross a road, unless there is a bridge." Such queries are of particular interest when users modify the database. Prior to making a change, they may want to get information what data they must provide, or after an unsuccessful attempt to update the database, they may want to find out more details about their --- or the system's --- failure.

5.2.3. Qualitative Answers.

SQL allows users to ask queries with quantitative results, i.e., about the values of tuples; however, it does not support queries with qualitative results, e.g., about the kind of relationship between objects. For instance, while it is possible to ask for "all roads in the district of Avila which are wider than the road from Madrid to Las Navas," it is impossible to formulate an SQL query for "the relation between the two roads from Madrid to Las Navas and from Las Navas to Avila," with an expected answer like "The road from Las Navas to Avila is narrower." Likewise, relationships cannot be used as variables, for instance to ask a query like "Find all scenic places which are in the same direction as El Escorial is from Las Navas."

The conceptual problem behind this shortcoming is that SQL---like any other relational query language --- is *first-order*, i.e., the predicates are not variable (Gallaire et al. 1984). GIS users frequently ask qualitative queries, e.g., "What is the directional relationship between Las Navas and El Escorial?"

5.2.4. 0bject Identity.

Since SQL is inherently value-based it provides only means to compare objects for the same value(s), but lacks the concept of identity. *Identity* is that property which distinguishes each object from all others (Khoshafian and Copeland 1986); therefore, object identity allows for the comparison whether two objects are the same, independent of their particular attribute values. Object identity has been recently acknowledged as a major component of next-generation database systems (Atkinson et al. 1989, Stonebraker et al. 1990), because it is crucial for any application in which objects change over time (Al-Taha and Barrera 1990). Geographic objects are prototypical examples thereof. A land parcel, for example, may change all its characteristics over time: the boundaries get resurveyed and adjusted, new owners buy the land, the land may be used differently, the parcel may get a new postal address, etc. Nevertheless, the piece of land is still the same and in order to relate its state at one time in history with another state it is necessary that each parcel has an object identifier which is independent of the descriptive values of the parcel.

6. Conclusion

The intent of this paper was to bring a new perspective to the ongoing discussion about the use of the SQL framework for the standard spatial query language. While SQL has been fairly successfully applied as a query language for standard database applications, such as banking accounts, its success in the domain of spatial applications, such as geographic information systems or CAD/CAM systems, has been very limited. Despite numerous attempts to extend SQL with various spatial features, no satisfying solutions for an SQL-based interactive spatial query language have been found.

A spatial SQL can be considered only a short-term "solution" for an interactive spatial query language. Extensions of the SQL framework with a spatial abstract data type (ADT), including spatial relationships as predicates, are possible; however, prior to discussing syntax extensions, the semantics of the operations upon the spatial ADT must be formally defined. Informal sets of operations to be carried out by a spatial database have been collected (Joseph and Cardenas 1988, Tomlin 1990); however, only little effort has been put into the formalization of spatial data models (Giffing 1988, Dorenbeck and Egenhofer 1991) and spatial relationships and operations such as cardinal directions (Peuquet 1988, Chang et al. 1989, Frank 1991b, Hernandez 1991) and topological relationships (Egenhofer 1989a, Egenhofer and Herring 1990, Egenhofer and Franzosa 1991).

Geographic databases need query languages which are more powerful and better suited than SQL or an extension of it. SQL *per se* is already difficult to use (Reisner 1981) and any addition to the SQL concepts increases its complexity. SQL lacks crucial concepts for a spatial query language, the most important deficit being the missing integration of data retrieval and representation of query results. The retrieval of data and its alphanumeric representation in a tabular format are insufficient for spatial data. Particularly the graphical display of query results in all its varieties cannot be handled with traditional query languages.

In lieu of a database query language, spatial applications need a database interface language, considering both the retrieval of data and the representation of query results. The design of such a language must start at the user level by investigating what kinds of operations users want to perform upon spatial databases and how they it (Egenhofer and Frank 1988b, Pizano et al. 1989, Mainguenaud and Portier 1990). Recent studies of the use of metaphors for the interaction with spatial data (Kuhn and Frank 1991) provide a starting point for investigations into database interface languages.

7. Acknowledgements

I appreciate Andrew Frank's and Robert Cicogna's comments on an earlier version of this paper.

References

Abel, D. (1988) "Relational Data Management Facilities for Spatial Information Systems", In D. Marble (ed.), Third International Symposium on Spatial Data Handling, Sydney, Australia, pp. 9-18.

Al-Taha, K. and Barrera, R. (1990) "Temporal Data and GIS: An Overview", in GISALIS '90, Anaheim, CA, pp. 244-254.

Allen, J. (1989) "Maintaining Knowledge about Temporal Intervals", Communications of the ACM, 26(11), 832-843.

ANSI (1986) "X3112-86-2 American National Standard Database Language SQL", American National Standards Institute.

ANSI-ISO Committee (1989) "Working Draft, Database Languages SQL 2 and SQL 3", American National Standards Institute and International Standards Organization.

Ariav, G. (1986) "A Temporally Oriented Data Model", ACM Transactions on Database Systems 11(4), 499-527.

Atkinson, M., Bancilhon, F., DeWitt, D., Dittrich, K., Maier, D., and Zdonik, S. (1989) "The Object-Oriented Database System Manifesto", in Deductive and ObjectOriented Databases, Kyoto, Japan.

Berman, R. and Stonebraker, M. (1977) "GEO-QUEL, A System for the Manipulation and Display of Geographic Data", ACM Computer Graphics, 11(2), 186-191.

Bertin, J. (1983) Semiology of Graphics, The University of Wisconsin Press, Madison, W1.

Chamberlin, D., Astrahan, M., Eswaran, K., Griffiths, P., Lorie, R., Mehl, J., Reisner, P., and Wade, B. (1976) "SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control", IBM Journal of Research and Development 20(6), 560-575.

Chang, S.K., Jungert, E., and Li, Y. (1989) "The Design of Pictorial Databases Based Upon the Theory of Symbolic Projections", in A. Buchmann, 0. Giinther, T. Smith, and Y. Wang (eds.) Symposium on the Design and Implementation of Large Spatial Databases, Lecture Notes in Computer Science 409, SpringerVerlag, New York, NY, pp. 303-323.

Codd, E. (1970) "A Relational Model for Large Shared Data Banks", Communications of the ACM 13(6), 377-387.

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

Date, C. (1988) "Defining Data Types in a Database Language", SIGMOD Record 17(2), 53-76.

Date, C. (1989b) A Guide to the SQL Standard, Addison-Wesley Publishing Company, Reading, MA.

Date, C. and White, C. (1989a) A Guide to SQL/DS, Addison-Wesley Publishing Company, Reading, MA.

Davis, E. (1986) Representing and Acquiring Geographic Knowledge, Research Notes in Artificial Intelligence, Morgan Kaufmann Publishers, Inc., Los Altos, CA.

Dorenbeck, C. and Egenhofer, M. (1991) "Optimization Strategies for Overlay Operations", in D. Mark and D. White (eds.), Autocarto 10, Baltimore, MD (in press).

Egenhofer, M. (1984) "Implementation of MAPQUERY, a Query Language for Land Information Systems" (in German), Technical Report 79, Institute for Geodesy and Photogrammetry, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Egenhofer, M. (1987) "An Extended SQL Syntax To Treat Spatial Objects", in Y.C. Lee (ed.), Second International Seminar on Trends and Concerns of Spatial Sciences, Fredericton, New Brunswick, Canada, pp. 83-95.

Egenhofer, M. (1988) "A Spatial SQL Dialect", Technical Report 100, Department of Surveying Engineering, University of Maine, Orono, ME.

Egenhofer, M. (1989a) "A Formal Definition of Binary Topological Relationships", in W. Litwin and H.-J. Schek (eds.), Third International Conference on Foundations of Data Organization and Algorithms (FODO), Paris, France, Lecture Notes in Computer Science 367, Springer-Verlag, New York, NY, pp. 457-472.

Egenhofer, M. (1989b) "Spatial Query Languages", PhD thesis, University of Maine, Orono, ME.

Egenhofer, M. (1990) "Interaction with Geographic Information Systems via Spatial Queries", Journal of Visual Languages and Computing 1(4) (in press).

Egenhofer, M. (1991) "Extending SQL for Cartographic Display", Cartography and Geographic Information Systems (in press).

Egenhofer, M. and Frank, A. (1988a) "Designing Object-Oriented Query Languages for GIS: Human Interface Aspects", in D. Marble (ed.), Third International Symposium on Spatial Data Handling, Sydney, Australia, pp. 79-96.

Egenhofer, M. and Frank, A. (1988b) "Towards a Spatial Query Language: User Interface Considerations", in D. DeWitt and F. Bancilhon (eds.), 14th International Conference on Very Large Data Bases, Los Angeles, CA, pp. 124133.

Egenhofer, M. and Frank, A. (1990) "LOBSTER: Combining Al and Database Techniques for GIS", Photogrammetric Engineering & Remote Sensing 56(6), 919-926.

Egenhofer, M. and Herring, J. (1990) "A Mathematical Framework for the Definition of Topological Relationships", in K. Brassel and H. Kishimoto (eds.), Fourth International Symposium on Spatial Data Handling, Zurich, Switzerland, pp. 814-819.

Egenhofer, M. and Franzosa, R. (1991) "Point-Set Topological Spatial Relations", International Journal for Geographical Information Systems, (in press).

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

Frank, A. (1984a) "Requirements for Database Systems Suitable to Manage Large Spatial Databases", in International Symposium on Spatial Data Handling, Zurich, Switzerland, pp. 38-60.

Gallaire, H., Minker, J., and Nicolas, J., (1984) "Logic and Databases: A Deductive Approach", ACM Computing Surveys 16(2), 153-185.

Go, A., Stonebraker, M., and Williams, C. (1975) "An Approach to Implementing a GeoData System", Technical Report, Memo ERL-M529, Electronics Research Laboratory, University of California, Berkeley, CA.

GUting, R. (1988) "Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems", in J. Schmidt, S. Ceri, and M. Missikoff (eds.), International Conference on Extending Database Technology, Venice, Italy, Lecture Notes in Computer Science 303, Springer-Verlag, New York, NY, pp. 506-527.

Hdrder, T. and Reuter, A. (1985) "Architecture of Database Systems for Non-Standard Applications" (in German), in A. Blaser and P. Pistor (eds.), Database Systems in Office, Engineering, and Science, Informatik Fachberichte 94, Springer Verlag, New York, NY, pp. 253-286.

Hernandez, D. (1991) "Relative Representation of Spatial Knowledge: The 2-D Case", in D. Mark and A. Frank (eds.) Cognitive and Linguistic Aspects of Geographic Space, Kluwer Academic Publishers, Dordrecht.

Herring, J., Larsen, R., and Shivakumar, J. (1988) "Extensions to the SQL Language to Support Spatial Analysis in a Topological Data Base", in GIS/LIS '88, San Antonio, TX, pp. 741-750.

Ingram, K. and Phillips, W. (1987) "Geographic Information Processing Using a SQLBased Query Language", in N.R. Chrisman (ed.), AUTO-CARTO 8, Eighth International Symposium on Computer-Assisted Cartography, Baltimore, MD, pp. 326-335.

Joseph, T. and Cardenas, A. (1988) "PICQUERY: A High Level Query Language for Pictorial Database Management", IEEE Transactions on Software Engineering 14(5), 630-638.

Khoshafian, S. and Copeland, G. (1986) "Object Identity", in N. Meyrowitz (ed.) ObjectOriented Programming Systems, Languages, and Applications --- OOPSLA '86, Portland, OR, SIGPLAN Notices 21(11), 406-416.

Korth, H. and Silberschatz, A. (1986) Database System Concepts, McGraw-Hill Book Company, New York, NY.

Kuhn, W. and Frank, A. (1991) "A Formalization of Metaphors and Image Schemas in User Interfaces", in D. Mark and A. Frank (eds.) Cognitive and Linguistic Aspects of Geographic Space, Kluwer Academic Publishers, Dordrecht.

Laurini, R. and Milleret, F. (1988) "Spatial Data Base Queries: Relational Algebra versus Computational Geometry", in M. Rafanelli, J. Klensin, and P. Svensson (eds.), Statistical and Scientific Database Management, 4th International Conference, Rome, Italy, Lecture Notes in Computer Science 339, SpringerVerlag, New York, NY, pp. 291-313.

Lorie, R. and Schek, H.-J. (1988) "On Dynamically Defined Complex Objects and SQL", in K.R. Dittrich (ed.), 2nd International Workshop on Object-Oriented Database Systems, Bad MOnster am Stein-Ebernburg, F.R. Germany, Lecture Notes in Computer Science 334, Springer-Verlag, New York, NY, pp. 323-328.

Mainguenaud, M. and Portier, M.-A. (1990) "Cigales: A Graphical Query Language for Geographical Information Systems", in K. Brassel and H. Kishimoto (eds.), Fourth International Symposium on Spatial Data Handling, Zurich, Switzerland, pp. 393-404.

Motro, A. and Yuan, Q. (1990) "Querying Database Knowledge", in H. Garcia-Molina and H. Jagadish (eds.), 1990 ACM SIGMOD, International Conference on Management of Data, Atlantic City, NJ, SIGMOD Record 19(2), 173-183.

Ooi, B.C., Sacks-Davis, R., and McDonell, K. (1989) "Extending a DBMS for Geographic Applications", in IEEE Fifth International Conference on Data Engineering, Los Angeles, CA, pp. 590-597.

Peuquet, D. (1988) "Toward the Definition and Use of Complex Spatial Relationships", in D. Marble (ed.), Third International Symposium on Spatial Data Handling, Sydney, Australia, pp. 211-223.

Pizano, A., Klinger, A., and Cardenas, A. (1989) "Specification of Spatial Integrity Constraints in Pictorial Databases", Computer, 22(12), 59-71.

Pullar, D. (1987) "Query Language for Spatial Relationships", in ASPRS-ACSM Annual Convention, Baltimore, MD, pp. 180-192.

Reisner, P. (1981) "Human Factors Studies of Database Query Languages: A Survey and Assessment", ACM Computing Surveys 13(l), 13-31.

Roussopoulos, N. and Leifker, D. (1985) "Direct Spatial Search on Pictorial Databases Using Packed R-Trees", in S. Navathe (ed.), International Conference on Management of Data, Austin, TX, SIGMOD Record 14(4), 17-31.

Roussopoulos, N., Faloutsos, C., and Sellis, T. (1988) "An Efficient Pictorial Database System for PSQL", IEEE Transactions on Software Engineering 14(5), 630-638.

Sarda, N. (1990) "Extensions to SQL for Historical Databases", IEEE Transactions on Knowledge and Data Engineering 2(2), 220-230.

Schek, H.-J. (1988) "Perspectives of Future Database Systems --- Extensibility and Obj ect- Orientation" (in German), in AM/FM International, European Division, Regional Conference, Siegen, F.R. Germany, pp. 2-14.

Sikeler, A. (1985) "Examination of Storage Structures for 3-Dimensional Objects" (in German), Technical Report, University Kaiserslautern.

Stonebraker, M., Rowe, L., Lindsay, B., Gray, J., Carey, M., Brodie, M., Bernstein, P., and Beech, D. (1990) "Third Generation Data Base System Manifesto", SIGMOD Record 19(3), 31-44.

Tomlin, C. (1990) Geographic Information Systems and Cartographic Modeling, Prentice-Hall, Englewood Cliffs, NJ.

Ullman, J. (1982) Principles of Database Systems, Computer Science Press, Rockville,
MD.

Ullman, J. (1988) Principles of Database and Knowledgebase Systems, Computer Science Press, Rockville, MD.

Waugh, T. and Healey, R. (1987) "The GEOVIEW Design: A Relational Database Approach to Geographical Data Handling", International Journal of Geographical Information Systems 1(2), 101-118.

Westlake, A. and Kleinschmidt, 1. (1990) "The Implementation of Area and Membership Retrievals in Point Geography using SQL", in Z. Michalewicz (ed.), Statistical and Scientific Database Management, Fifth International Conference, Charlotte, NC, Lecture Notes in Computer Science 420, SpringerVerlag, New York, NY, pp. 200-218.

A FORMALIZATION OF METAPHORS AND IMAGE-SCHEMAS IN USER INTERFACES

WERNER KUHN and ANDREW U. FRANK
National Center for Geographic Information and Analysis
and
Department of Surveying Engineering
University of Maine
Orono, ME 04469 (USA)
Kuhn@mecanl, Frank@mecanl

ABSTRACT. Sound engineering approaches to user interface design require the formalization of key interaction concepts, one of them being metaphor. Work on interface metaphors has, however, been largely non-formal so far. The few existing formal theories of metaphor have been developed in the context of natural language understanding, learning, or reasoning. We propose to formalize interface metaphors by algebraic specifications. This approach provides a comprehensive formalization for the essential aspects of metaphorical user interfaces. Specifically, metaphor domains are being formalized by algebras, metaphorical mappings by morphisms, and image-schemas by categories. The paper explains these concepts and the approach, using examples of spatial and spatializing metaphors.

1. Introduction

Metaphor pervades communication. Metaphorical thought, action, and language are not only essential to interpersonal communication [Lakoff and Johnson 1980], but to human-computer communication as well. Since communication with computers is an outcome of design, building effective user interfaces requires a formal theory of metaphor [Carroll, Mack, and Kellogg 19881.

This paper presents a methodological foundation for such a theory. It proposes a formalization of interface metaphors based on abstract algebra [MacLane and Birkhoff 1967]. The algebraic approach allows designers to specify not only metaphor domains, but metaphorical mappings and image-schemas as well. The paper describes the basic elements of the approach.

After explaining the underlying understanding of interface metaphors in Section 2, important related work is discussed in Section 3. The formalization method is presented with a running example in Section 4 and applied to spatial zooming operations in Section 5. The last section contains conclusions and suggestions for future work.

2. Interface Metaphors and Image-Schemas

Systems with user interfaces based on the DESKTOP metaphor and its variants [Apple Computer 1987, Smith et al. 1982] have established metaphor as a powerful means to control complexity in human-computer interaction [Carroll, Mack, and Kellogg 19881. The structure and role of interface metaphors are, however, still understood in many different ways [Blumenthal 1990, Carroll, Mack, and Kellogg 1988, Erickson 1990a, Smith et al. 1982, Wozny 1989].

The understanding of interface metaphors underlying our formalization derives from recent work in cognitive linguistics [Johnson 1987b, Lakoff 1987, Lakoff and Johnson 1980]. Johnson [1987b, p. XIV] has characterized metaphor as *"...a pervasive mode of understanding by which we project patterns from one domain of experience in order to structure another domain of a different kind*." The two domains are commonly called the *source* and *target* domains of a metaphor and the metaphorical projection is a mapping from source to target.

Interface metaphors, in this projective view, go beyond explaining unfamiliar domains to novices. They determine how labor is distributed between a user and a system, what concepts a user has to deal with, and in what terms user and system communicate. In short, they structure application domains and organize tasks.

It is not clear how much metaphor-independent structure there can be in a target domain. A recent case study of metaphors in interface design [Erickson 1990a] indicates that the structure available to the designer before a metaphor is consciously chosen is either quite abstract or already metaphorical. In Erickson's example of a DESKTOP metaphor extension, that structure consists of links between documents. Interestingly, the link structure is a so-called *image-schema*. Image-schemas are image-like reasoning patterns, consisting of a small number of parts and relations, made meaningful by sensori-motor experience [Johnson 1987b, Lakoff and Johnson 19801. Examples are the CONTAINER, SURFACE, LINK, PATH, NEAR-FAR, PART-WHOLE, and CENTER-PERIPHERY schemas.

Image-schemas may in fact be the kind of structure which is preserved by interface metaphors. This assumption agrees with Lakoffs invariance hypothesis [Lakoff 1990] which claims that image-schemas remain invariant under metaphorical mappings. It also generalizes to user interfaces of any kind the suggestion that image-schemas play a fundamental role in interfaces of Geographic Information Systems [Mark 1989]. The design of interfaces for international use [Nielsen 1990] is one area where image-schemas, due to their presumed culture-independence, are likely to be of particular importance.

3. Existing Metaphor Formalizations

While there is a vast literature on metaphor and analogy (e.g., [Johnson 1981, Layoff and Johnson 1980, Mac Cormac 1985, Ortony 1979, Vosniadou and Ortony 1989]), formal approaches are rare (see [Hall 1989] for a survey and [Indurkhya 1989, Martin 19901 for recent treatments). Most existing formalizations employ traditional knowledge representation formalisms, such as semantic networks, to specify metaphorical domains, but not mappings or image-schemas. Furthermore, the formalizations have been developed for natural language understanding, machine learning or automated reasoning, while discussions of metaphor in human-computer interaction have been largely non-formal so far.

The best known formal theory of metaphor and analogy is Gentner's structure mapping theory [Gentner 1983, 19891. It describes analogies as mappings between source and target domains, each represented by semantic networks. It does not formalize the mappings themselves, however, and rests on a syntactical distinction of different kinds of relations. While Gentner's theory deals with structural aspects, it neglects the role of tasks in metaphor use [Carroll, Mack, and Kellogg 19881.

Our formalization addresses these problems by formalizing mappings as morphisms and expressing tasks and actions through algebraic operators and their effects. Describing domains algebraically rather than relationally may only be a syntactic difference; it does, however, allow for relating metaphors to tasks. For example, the informal task description of parts of the DESKTOP metaphor given in [Carroll, Mack, and Kellogg 19881 can directly be expressed by an algebraic formalism (see next section).

In the terminology of Carroll et al., our approach is primarily *structural*, addressing the representation of metaphor domains and mappings, but incorporates the *pragmatic* aspects of actual user tasks. It does not deal with operational analyses of the effects of interface metaphors on learning or using a system.

Most theories of metaphor stress the importance of mapping source structure into the target domain. Surprisingly, the mathematical devices for describing structure mappings, i.e., morphisms, are rarely applied. An exception is a forthcoming treatment [Indurkhya 19891 which describes cognitive models by algebras and metaphorical projections by morphisms.

Our independently developed algebraic approach differs from Indurkhya's theory mainly by:

- describing metaphorical domains as elements of algebra *classes*, thereby dealing with the relation of metaphors to image-schemas;

- specifically addressing *interface* metaphors;

- using the established software design technique of algebraic specifications.

Existing metaphor formalizations provide little support for interface designers because they are either purely theoretical or rely on specialized software in prototype form. The advantage of using the technique of algebraic specifications is that it allows designers to apply the method immediately, even with the possibility of using computer supported design tools such as Larch [Guttag, Homing, and Wing 1985] or CLEAR [Burstall and Goguen 19811.

4. An Algebraic Approach

4.1. ALGEBRAIC SPECIFICATIONS

Algebraic specifications describe objects in terms of operations. The associated theory has been developed and used for specifying abstract data types [Goguen, Thatcher, and Wagner 1978, Guttag and Homing 19781; practical introductions can be found in [Ehrig and Mahr 1985, Liskov and Guttag 1986, Woodcock and Loomes 19891.

An algebraic specification consists of three parts:

- a list of *sorts*, naming the objects involved;

- a list of *operators* (Ops), giving the sorts of arguments and results;

- a list of *axioms* (Eqs), defining equivalences for different sequences of operations.

Sorts and operators together are called the *signature* of an algebraic specification. *Algebras* are interpretations of signatures, assigning a set to each sort and an operation on these sets to each operator. An algebraic specification, i.e., sorts, operators, and axioms together, describes a *category*, i.e. the class of algebras satisfying the specification [Ehrich, Gogolla, and Lipeck 1989].

The technique of algebraic specifications is best demonstrated by applying it immediately to the formalization of source and target domains from Apple's composite DESKTOP metaphor [Apple Computer 19871.

4.2. SOURCE AND TARGET DOMAINS

The following is an algebraic specification of physical office desktops. It lists three sorts: Desktops, Folders, and Bools (truth values). Desktops are the sort actually being specified, Folders will be specified below, and Bools are assumed to have been specified previously. There are operators to create a new desktop, put a folder on a desktop, get a folder from a desktop, and check whether a folder is on a desktop. The meaning of these operators is defined by the equations and does not rely on an interpretation of their names. The equations make free use of universally quantified sort variables (dt for desktops, f for folders, etc.).

```
Desktop
Sorts           Desktop, Folder, Bool
Ops     new:    Æ Desktop
        put:    Desktop x Folder Æ Desktop
        get:    Desktop x Folder Æ Desktop
        on:     Desktop x Folder Æ Bool
Eqs     on(new,f) = false
        on(put(dt,fl),f2)   if fl=f2 then true else on(dt,f2)
        get(put(dt,fl),f2)  if fl=f2 then dt else put(get(dt,f2),fl)
```

Now, the technique is applied to specify the corresponding target domain in Apple's DESKTOP metaphor. The sort and variable names for the electronic analogues of office objects are prefixed by "El" and "el", respectively.

```
ElDesktop
Sorts           ElDesktop, ElFolder, Bool
Ops     new:    Æ ElDesktop
        put:    ElDesktop x ElFolder Æ ElDesktop
        get:    ElDesktop x ElFolder Æ ElDesktop
        on:     ElDesktop x ElFolder Æ Bool
Eqs     on(new,elf) = false
        on(put(eldt,elfl),elf2)      if elfl=elf2 then true else on(eldt,elf2)
        get(put(eldt,elfl),elf2)     if elfl=elf2 then eldt else
        put(get(eldt,elf2),elf 1)
```

Obviously, these two specifications are identical up to sort and variable names. In practice, this means that Macintosh desktops behave like real desktops with respect to the operations defined. For the sake of simplicity, only the most salient features have been specified. By including additional operators, dissimilarities could be shown, such as the fact that things can fall from physical, but not from electronic desktops. In an actual interface design process, the designer has to decide which features of a source domain are to be considered salient and which are not.

Next, clipboards are specified. First the physical variety, with the assumption that clips are added to and removed from the top. Note that this specification resembles that of desktops. We will come back to this parallel when discussing image-schemas in 4.4.

```
Clipboard
Sorts           Board, Clip, Bool
Ops     new:    Æ Board
        put:    Board x Clip Æ Board
        get:    Board )Æ Board
        on:     Board x Clip )Æ Bool
Eqs     on(new,c) = false
        on(put(b,cl),c2) = if cl=c2 then true else on(b,c2)
        get(put(b,c)) = b
```

Secondly, the electronic variety of clipboards, as provided by the Macintosh to move data between documents, is specified:

```
ElClipboard
Sorts           ElBoard, ElClip, Bool
Ops     new:    Æ ElBoard
        put:    ElBoard x ElClip Æ ElBoard
        get:    ElBoard  Æ ElBoard
        on:     ElBoard x ElClip )Æ Bool
Eqs     on(new,elc) = false
        on(put(elb,elcl),elc2) = if elcl=elc2 then true else false
        get(put(elb,elc)) =  put(elb.elc)
```

The axioms show two differences between physical and electronic clipboards:

• a clip remains on the Macintosh clipboard only until the next clip is put on -then it is lost;

• getting back a clip does not change the electronic clipboard and the same clip can be retrieved multiple times.

While the first mismatch may be harmful, the second is rather an advantage. The point is, however, that the specification *reveals* these mismatches. In an actual interface design process, it is the designer's task to decide about their desirability [Halasz and Moran 1981, Johnson 1987a].

Folders are another part of the DESKTOP metaphor. Here is a specification of the salient features of physical office folders:

```
Folder
Sorts           Folder, Document, Name, Bool
Ops     new:    Æ Folder
        label:  Folder x Name Æ Folder
        insert: Folder x Document Æ Folder
        remove: Folder x Document )Æ Folder
        empty:  Folder Æ Folder
        in: Folder x Document Æ Bool
Eqs     remove(label(f,n),d) = label(remove(f,d),n)
        remove(insert(f,dl),d2) = if dl=d2 then f else
        insert(remove(f,d2),dl)
        empty(new) = new
        empty (I abel (f,n)) = label(empty(f),n)
        empty (insert(f, d)) = empty(f)
        in(new,d) = false
        in(labelff,n),d) = in(f,d)
        in(insert(f,d 1),d2) = if dl=d2 then true else in(f,d2)
```

The following specification of electronic folders is very similar. The only difference is that the empty operator is missing because the Macintosh does not offer such an operation. (It does, of course, allow users to select all documents in a folder and remove them or to discard an entire folder, but these are different operations.)

```
ElFolder
Sorts           ElFolder, ElDocument, Name, Bool
Ops     new:    Æ ElFolder
        label:  ElFolder x Name Æ ElFolder
        insert: ElFolder x ElDocument )Æ ElFolder
        remove: ElFolder x ElDocument Æ ElFolder
        in:     ElFolder x ElDocument Æ Bool
Eqs     remove(label(elf,n),eld) = label(remove(elf,eld),n)
        remove (insert (elf,el d 1 ),eld2) = if eldl=eld2 then elf else
        insert (remove (el f,eld2), eld 1)
        in(new,eld) = false
        in(label(elf,n),eld) = in(elf,eld)
```

in(insert(elf,eldl),eld2) = if eldl=eld2 then true else in(elf,eld2)

The final pair of DESKTOP metaphor domains to be specified here is that of trash cans. Some operators (e.g., the possibility to discard folders) have been omitted for brevity:

Trash Can
Sorts Trash, Document, Bool
Ops new: Æ Trash
discard: Trash x Document Æ Trash
remove: Trash x Document Æ Trash
empty: Trash Æ Trash
in: Trash x Document Æ Bool
Eqs remove(discard(t,dl),d2) = if dl=d2 then t else
discard(remove(t,d2),dl)
empty(new) = new
empty (disc ard(t,d)) = new
in(new,d) = false
in(discard(t,dl),d2) = if dl=d2 then true else in(t,d2)

The specification of the target domain is similar to that of the source domain; however, it also reveals a mismatch:

ElTrash
Sorts ElTrash, ElDoc, Bool
Ops new: Æ ElTrash
discard: ElTrash x ElDoc Æ ElTrash
remove: ElTrash x ElDoc Æ ElTrash
empty: ElTrash Æ ElTrash
in: ElTrash x ElDoc Æ Bool
Eqs remove(discard(et,edl),ed2) = if edl=ed2 then et else
discard(remove(et,ed2),edl)
empty(new) = new
empty (discard (et,ed)) = new
in(new,ed) = false
in(discard(et,edl),ed2) = <u>true or false</u>

The difference in the last equation captures the fact that a Macintosh user does not have complete control over the Trash can contents. A document can be retrieved immediately after discarding it, but may be lost after another operation, such as opening a document or inserting a disk.

Each specification of a source or target domain given in this section describes a class of algebras. The *designer's* conceptual model of a domain is seen as an algebra in this class, L e., an interpretation of the corresponding specification which assigns symbols to the sorts and operators such that the axioms hold. An important challenge in implementing an interface metaphor is to make sure that the *user's* conceptual model belongs to the same class.

4.3. METAPHOR MAPPINGS

Metaphors are mappings from source to target domains. When the domains are specified algebraically, it is natural to use morphisms, which are mappings between algebras, to define metaphorical mappings.

There are different kinds of morphisms, reflecting different degrees of similarities between algebras. The weakest kind of similarity is a *signature-morphism* which is a correspondence between the signatures defining two algebras [Ehrich, Gogolla, and Lipeck 1989]. A stronger kind is a *homomorphism* between two algebras A and B. It is a family of mappings from the sets of A to those of B, such that it does not matter whether an operator is first applied in A and the result mapped to B, or the arguments are first mapped to B and the corresponding operator in B is applied. Thus, homomorphisms are structure-preserving mappings. A one-to-one homomorphism is called an *isomorphism*.

If the source and target domain are both specified, as in the above examples, the morphism is determined. It indicates the degree of similarity between source and target for the specified (and, in this case,' implemented) interface metaphors:

- The mapping between office and electronic *desktops* is an isomorphism, since the effects of the specified operators on office desktops correspond to the effects of the analogous operators on electronic desktops, and vice versa.

- The mapping from physical to electronic *clipboards* is a signaturemorphism. The signatures are the same, but the effects of some of the operators are different. For example, the signatures of the physical put and get operators map to those of the electronic ones, but the effects of applying get after put are different, as can be seen from the third axioms in the two specifications.

- The *folder* specifications exemplify another isomorphism, but only with respect to the new, label, insert, remove, and in operators; the empty operator has no equivalent in the target domain.

- The *trash can* specifications are linked by another signature-morphism. The signatures correspond, but the result of the in operator after discarding a document can be different for the two domains.

In the normal course of an interface design, the target domains are not predetermined. Source domains are specified first and appropriate morphisms are chosen to achieve the intended matches. These morphisms then determine parts of the target domain by projecting the desired elements of the source structure.

4.4. IMAGE-SCHEMAS

The algebraic specifications of source and target domains from the DESKTOP metaphor exhibit two image-schemas: Desktops and clipboards are instantiations of the SURFACE schema; folders and trash cans represent the CONTAINER schema. The SURFACE schema embodies the logic of a surface on which items can be put. The CONTAINER schema consists of an interior, an exterior, and a boundary between them, so that items can be in a container or outside of it [Lakoff 1987].

Thus, not only are image-schematic structures believed to be invariant in metaphorical mappings, they can also be common to different parts of a composite metaphor. Composite metaphors whose domains are related to common imageschemas are more likely to be perceived as *coherent* than those where this is not the case. It can be argued [Erickson 1990b] that the success of the Macintosh DESKTOP metaphor is in part due to a largely consistent use of the CONTAINER image-schema.

If an image-schema is invariant in a metaphorical mapping, it must be a common part of the source and target domains. It is therefore possible to obtain the algebraic specifications of these domains by extending a common core specification which formalizes an image-schema. Such a process of adding operators to algebraic specifications is called an *enrichment* [Ehrich, Gogolla, and Lipeck 1989]. Thus, a formal version of Lakoff's invariance hypothesis is: *For any metaphor, there is an algebraic specification which describes an image-schema or a combination of imageschemas and which can be enriched toward specifications of the source and target domains*. Since an algebraic specification describes a class of algebras or category, image-schemas are formalized as categories. These categories contain the algebras of the source and target domains as well as the morphisms between them.

The following is an algebraic specification of the SURFACE image-schema. It can easily be enriched toward the above specifications of the physical and electronic desktops by renaming the sorts (Surface Æ Desktop, Item Æ Folder, etc.) and adding the get operator with its axiom. The same can be said for the physical, but not for the electronic clipboard: The fact that a previously clipped item is lost when a new item is put on is not only a deviation from the behavior of physical clipboards, but also from the logic of the SURFACE schema.

```
Surface
Sorts           Surface, Item, Bool
Ops     new:    Æ Surface
        put:    Surface x Item Æ Surface
        on:     Surface x Item Æ Bool
Eqs     on(new,i) = false
        on(put(s,il),i2) = if il=i2 then true else on(s,i2)
```

The other image-schema dominating the DESKTOP metaphor is the CONTAINER. Its specification happens to be isomorphic to that of the SURFACE schema. This correspondence may, in fact, contribute to the coherence of the DESKTOP metaphor. From a cognitive perspective, however, two different kinds of experiences are described. The CONTAINER schema can be specified as follows (see also [Guttag, Homing, and Wing 19851):

```
Container
Sorts           Container, Item, Bool
Ops     new:    Æ Container
        insert: Container x Item Æ Container
        in:     Container x Item Æ Bool
Eqs     in(new,i) = false
        in(insert(c,i1),i2) = if il=i2 then true else in(c,i2)
```

This specification can again be enriched toward the previously given specifications of physical and electronic folders as well as physical trash cans. Thus, all these objects behave the way one intuitively expects from containers. No enrichment is possible, however, for electronic trash cans: The second axiom of the CONTAINER specification is violated by those cases where a discarded document is not in the trash without having been removed. Given how deeply rooted image-schemas are in human cognition, such mismatches with image-schemas are more likely to be harmful than source-target differences which are above the image-schematic level.

5. A Spatial Application

5. 1. SPATIALIZED AND SPATIAL DOMAINS

The DESKTOP metaphor affords a spatialization for an abstract, non-spatial application domain: The source concepts of the physical office impose a spatial structure on the abstract target domain of information processing. A wide range of other spatializing interface metaphors provides evidence for the crucial role of spatialization in human-computer interaction. Examples include the NAVIGATION [Conklin 1987], TOURIST [Fairchild, Meredith, and Wexelblatt 1989], ROOM [Henderson and Card 19861, and MUSEUM [Travers 1989] metaphors. Indeed, the two- or three-dimensional nature of visual interaction itself implies a spatialization for visual interfaces [Tauber 1987] in general.

At the same time, cognitive science has established that "most of our fundamental concepts are organized in terms of one or more spatialization metaphors" [Lakoff and Johnson 1980] and that many image-schemas are spatial in nature. These findings suggest, too, that spatialization is probably the rule rather than the exception for interface metaphors.

When dealing with geographic space, however, interface metaphors for inherently spatial applications are of primary interest. An example of such an interface metaphor is furnished by zoom and pan operations in Geographic Information Systems (GIS) and other graphics applications.

The terms "zoom" and "pan" are used in computer graphics in analogy to manipulations of cameras or other optical instruments. An investigation of zoom and pan interface designs for GIS concluded, however, that intuitive and effective interface tools require a deeper understanding of the metaphors involved [Jackson 1990]. The remainder of this section applies the formal tools developed in the previous section to an analysis of the metaphorical and image-schematic structure of zooming.

5.2. ZOOMING AND THE "DISPLAYS ARE VIEWS" METAPHOR

The Oxford English Dictionary (second edition, 1989) defines the original meaning of "zoom" as follows: "*To make a continuous low-pitched humming or buzzing sound; to travel or move (as if) with a ’zooming’ sound; to move at speed, to hurry*."

The use of the term in photography and cinematography is, thus, already doubly metaphorical: It explains the variation of the focal length by a (fictive) motion of rapidly closing up on a subject which, in turn, is metaphorically related to the corresponding sound effect. (It is an interesting twist that one of the key metaphors in visual interfaces is originally echoic, i.e. rooted in auditory perception.)

Combined with our deeply rooted visual experience that the viewing distance determines *what* we see, zooming acquires the interpretation of changing the level of detail at which we perceive and conceptualize the world or a computer model.

This understanding of zooming suggests the more general metaphor DISPLAYS ARE VIEWS, where views are understood as visual fields, containing what humans see in a given situation [Kuhn 1990]. This metaphor accommodates various transformations of the visual field in a common framework: Zooming is the transformation caused by changing the viewing distance; panning is moving the view to another part of a it panorama" without changing the level of perceived detail. Since transformations of the visual field correspond to "basic cognitive processes such as focusing, scanning, superimposition, figure-ground shifting, vantage-point shifting" [Lakoff 1988], they are ideal candidates for metaphor sources.

5.3. IMAGE-SCHEMAS IN "DISPLAYS ARE VIEWS"

The basic image-schema involved in the visual field is the CONTAINER schema. It structures the visual field as a bounded space, consisting of a boundary, an interior, and an exterior: Things are either in or out of sight and they come into or go out of it [Lakoff 1987]. The visual field has also a center of attention and a surrounding region. Thus, a CENTER-PERIPHERY schema is combined with the CONTAINER schema, providing for the distinction of foveal and peripheral vision.

Enriching the previously specified CONTAINER schema by a center-periphery distinction produces the following specification (at this point, no axioms are given to constrain the inCenter operator):

```
Container with Center
Sorts             CentCont, Item, Bool
Ops     new:      Æ CentCont
        insert:   CentCont x Item Æ CentCont
        in:       CentCont x Item  Æ Bool
        inCenter:           CentCont x Item Æ Bool
Eqs     in(new,i) = false
        in(insert(cc,il),i2) = if il=i2 then true else in(cc,i2)
```

The visual field is further characterized by an interaction of the PART-WHOLE with the NEAR-FAR schema. We experience objects in the world as configurations of parts, forming wholes. Our perception has evolved so that it can distinguish parts and wholes. Unaided visual perception, in particular, relies on motion of the body or of the objects to extend this distinction beyond the limited range of configurations present in one view: Getting parts into view involves moving nearer and vice versa. This connection is the essence of zooming.

To formalize the superimposition of the PART-WHOLE and NEAR-FAR schemas on top of the centered CONTAINER, the above specification is further enriched by a closeUp operator and a part-whole relation (which actually belongs to a separate Item specification). Let us call this multiply enriched container a View Container. With these additional operators, it becomes possible to capture the meaning of zooming in one axiom for the closeUp operator:

```
View Container
Sorts             ViewCont, Item, Bool
Ops     new:      Æ ViewCont
        insert:   ViewCont x Item Æ ViewCont
        in:       ViewCont x Item Æ Bool
        inCenter:           ViewCont x Item Æ Bool
        part0f:   Item x Item Æ Bool
        closeUp:ViewCont Æ ViewCont
Eqs     in(new,i) = false
        in(insert(vc,il),i2)=if il=i2 then true else in(vc,i2)
        in(closeUp(vc),i) inCenter(vc,i) or (inCenter(vc,il) and part0f(i,il))
```

Note that the closeUp operator is specified quite generally: It is only constrained to keep center items in view and get their parts into view; no details about the degree of zooming or the extent of the center region are given. More specific axioms would require metric concepts, while an image-schematic analysis is only concerned with topological properties.

Further enrichments with metric concepts would lead to complete source (view) and target (display) specifications for a particular user interface. They could involve, for example, a viewing distance in the source domain and a zoom factor or view extent in the target domain.

It should be emphasized that DISPLAYS ARE VIEWS is a metaphor and not a literal equivalence. The matching between the two domains of a metaphor is by definition partial. Aspects which are not image-schematic, such as the shape of the container boundary (elliptic, rectangular) or the type of optical projection involved (central, parallel), need not be equal or even comparable.

6. Conclusions

An algebraic approach to the formalization of interface metaphors has been presented. Algebraic specifications define metaphor domains as algebras, metaphorical mappings as morphisms, and invariant image-schemas as categories. The gist of the approach should be captured in the examples from a spatialization metaphor (DESKTOP). The sketched application to zoom operations should demonstrate the relevance to the design of user interfaces for systems dealing with spatial information.

The proposed formalization of interface metaphors is not a formal method for the design of metaphorical interfaces, but we believe it is a useful and immediately applicable step toward it. The principal strengths of the proposal are that:

- the underlying understanding of metaphors is in agreement with recent findings in cognitive science;
- the algebraic approach provides a comprehensive formalization of all essential aspects of metaphors: conceptual domains, structure-mappings, and image-schemas;
- the technique of algebraic specifications is well established in software design and supported by design tools.

The proposed formalization has yet to be validated in actual user interface designs. Our initial experiences from applications to Geographic Information Systems [Jackson 1990] and to an interface based on book metaphors [Volta, Cicogna, and Kuhn 1990] have been positive. The rigor imposed and the questions raised by the approach have clearly been beneficial in the design process.

Formal approaches to design are mostly motivated by the need to evaluate designs. Distinguishing different kinds of morphisms and separating the image-schematic structure from surface-level structure are solid bases for such evaluations, but much more theoretical and experimental work needs to be done in this direction. Evaluating interface metaphors also involves dealing with the coherence of composite metaphors. Since image-schemas are believed to play an important role for coherence, their formalization is a key feature of this approach.

Some aspects of our work may have implications beyond human-computer interaction. The description of metaphorical domains by algebraic specifications, for instance, is an attempt at formalizing the notion of Idealized Cognitive Models (ICM) [Lakoff 1987]. It accommodates propositional and image-schematic structure as well as metaphoric mappings. Another general result is the formal version of the invariance hypothesis.

7. Acknowledgements

Support from NSF under NCGIA grant No. SES 88-10917, from Digital Equipment Corporation under grant No. TP-765536, and from Intergraph Corporation is gratefully acknowledged. We also thank Max Egenhofer and Jeffrey Jackson for discussions on the topic.

References

Apple Computer, Inc. 1987. Human Interface Guidelines: The Apple Desktop Interface. Reading, MA: Addison-Wesley.

Blumenthal, B. 1990. Incorporating Metaphor in Automated Interface Design. Proceedings Interact '90 : 659-664.

Burstall, R.M., and J.A. Goguen. 1981. An Informal Introduction to Formal Specifications Using CLEAR. In The Correctness Problem in Computer Science. Edited by R. S. Boyer and J. Moore. 185-213. New York: Academic Press.

Carroll, J.M., R.L. Mack, and W.A. Kellogg. 1988. Interface Metaphors and User Interface Design. In Handbook of Human-Computer Interaction. Edited by M. Helander. 67-85. North-Holland: Elsevier Science Publishers.

Conklin, J. 1987. Hypertext: An Introduction and Survey. IEEE Computer 20 (9) : 17-41.

Ehrich, H.-D., M. Gogolla, and U.W. Lipeck. 1989. Algebraische Spezifikation abstrakter Datentypen. Leitfaden und Monographien der Informatik. Stuttgart: B.G. Teubner.

Ehrig, H., and B. Mahr. 1985. Fundamentals of Algebraic Specification. Berlin: Springer-Verlag.

Erickson, T.D. 1990a. Working with Interface Metaphors. In The Art of HumanComputer Interface Design. Edited by B. Laurel. 65-73. Addison-Wesley.

Erickson, T.D. 1990b. Personal communication.

Fairchild, K., G. Meredith, and A. Wexelblatt. 1989. The Tourist Artificial Reality. Proceedings ACM CHI89 : 299-304.

Gentner, D. 1983. S tructure- Mapping: A Theoretical Framework for Analogy. Cognitive Science 7 : 155-170.

Gentner, D. 1989. The mechanisms of analogical learning. In Similarity and Analogical Reasoning. Edited by S. Vosniadou and A. Ortony. 199-241. Cambridge, UK: Cambridge University Press.

Goguen, J. A., J. W. Thatcher, and E. G. Wagner. 1978. An Initial Algebraic Approach to the Specification, Correctness, and Implementation of Abstract Data Types. In Current Trends in Programming Methodology. Edited by R. Yeh. 80-149. Englewood Cliffs, N.J.: Prentice-Hall.

Guttag, J.V., and J.J. Homing. 1978. The Algebraic Specification of Abstract Data Types. Acta Informatica 10 (1) : 27-52.

Guttag, J.V., J.J. Homing, and J.M. Wing. 1985. Larch in Five Easy Pieces. Digital Equipment Corporation, Systems Research Center.

Halasz, F., and T.P. Moran. 1981. Analogy Considered Harmful. International Journal of Man-Machine Studies 14 : 383-386.

Hall, R.P. 1989. Computational Approaches to Analogical Reasoning: A Comparative Analysis. Artificial Intelligence 39 : 39-120.

Henderson, D.A., Jr., and S.K. Card. 1986. Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface. ACM Transactions on Graphics 5 (3) : 211-243.

Indurkhya, B. 1989. Metaphor and Cognition. Manuscript.

Jackson, J.P. 1990. Visualization of Metaphors for Interaction With Geographic Information Systems. Masters of Science Thesis, University of Maine, Department of Surveying Engineering.

Johnson, J. 1987a. How faithfully should the electronic office simulate the real one? ACM SIGCHI Bulletin 19 (2) :21-25.

Johnson, M. 1981. Metaphor in the Philosophical Tradition. In Philosophical Perspectives on Metaphor. Edited by M. Johnson. 3-47. Minneapolis: University of Minnesota Press.

Johnson, M. 1987b. The Body in the Mind. Chicago: The University of Chicago Press.

Kuhn, W. 1990. Are Displays Maps or Views? Technical Report, Department of Surveying Engineering, University of Maine.

Lakoff, G. 1987. Women, Fire, and Dangerous Things. Chicago: The University of Chicago Press.

Lakoff, G. 1988. Cognitive Semantics. In Meaning and Mental Representations. Edited by U. Eco, M. Santambrogio and P. Violi. 119-154. Bloomington: Indiana University Press.

Lakoff, G. 1990. The Invariance Hypothesis: is abstract reason based on imageschemas? Cognitive Linguistics 1 (1) : 39-74.

Lakoff, G., and M. Johnson. 1980. Metaphors We Live By. Chicago: The University of Chicago Press.

Liskov, B., and J. Guttag. 1986. Abstraction and Specification in Program Development. The MIT Electrical Engineering and Computer Science Series. Cambridge, MA: The MIT Press (MacGraw-Hill).

Mac Cormac, E.R. 1985. A Cognitive Theory of Metaphor. Cambridge, MA: The MIT Press.

MacLane, S., and G. Birkhoff. 1967. Algebra. New York: Macmillan.

Mark, D. 1989. Cognitive Image-Schemata for Geographic Information: Relations to User Views and GIS Interfaces. Proceedings GISILIS'89 2 : 551-560.

Martin, J.H. 1990. A Computational Model of Metaphor Interpretation. New York: Academic Press.

Nielsen, J. 1990. Designing User Interfaces for International Use. Amsterdam: Elsevier Science Publishers.

Ortony, A. 1979. Metaphor and Thought. Cambridge, England: Cambridge University Press.

Smith, D.C.S., C. Irby, R. Kimball, B. Verplank, and E. Harslam. 1982. Designing the Star User Interface. BYTE. 7 (4) April 1982 : 242-282.

Tauber, M.J. 1987. On Visual Interfaces and their Conceptual Analysis. In Visualization in Programming. Edited by P. Gorny and M.J.Tauber. 106-123. Springer-Verlag.

Travers, M. 1989. A Visual Representation for Knowledge Structures. Proceedings Hypertext '89 : 147-158.

Volta, G.S., R. Cicogna, and W. Kuhn. 1990. The MaineBook. Technical Report, Department of Surveying Engineering, University of Maine.

Vosniadou, S., and A. Ortony. 1989. Similarity and Analogical Reasoning. Cambridge: Cambridge University Press.

Woodcock, J., and M. Loomes. 1989. Software Engineering Mathematics. SEI Series in Software Engineering. Reading, MA: Addison-Wesley.

Wozny, L.A. 1989. The application of metaphor, analogy, and conceptual models in computer science. Interacting with Computers 1 (3) : 273-283.

MATCHING REPRESENTATIONS OF GEOGRAPHIC LOCATIONS

Matthew McGranaghan
National Center for Geographic Information and Analysis
University of Maine
Orono, ME 04469 (USA)
matt@ mecan1.bitnet
and
Geography Department
University of Hawaii
Honolulu, HI 96822

ABSTRACT

Many references to geographic locations are made linguistically, using words to describe the location in terms of objects in the landscape and the relations among them. In order to attach coordinates to these locations, one must be able to identify them in a database which contains representations of the objects and relations as well as coordinates for these features. This article assesses the correspondence between the features used in describing locations in an herbarium collection and those represented in standard cartographic databases. An alternative representation for geographic data is suggested.

1. Introduction

> “…in general, there is a basic level of specificity of description. With objects, this is Rosch's "basic level"; there seems to be the equivalent of a basic level for spatial relations in scenes, but since scenes and situations are much more difficult to circumscribe than objects, their categorization properties have not been studied." (Herskovits 1986, 113)

This chapter is concerned with automating the conversion of descriptions of geographic locations to systematic coordinates. This facility would speed the integration of a large volume of non-coordinate- referenced spatial data, such as is now held in natural history collections, with geo-referenced data sets. The task is essentially one of matching geographic patterns found in the descriptions with those recorded in a geographic database. While this may seem to involve machines "understanding" descriptions, and indeed, it is convenient to use that metaphor, Winograd and Flores (1986) argue convincingly that computers, are symbol processors, and that any illusion that they "understand" is due to sufficiently manipulable symbols retaining meaning for humans. The importance of representation is clear.

Interpreting a description of a location requires establishing correspondence between two representations of the Earth's surface. One representation is the database of the world, which presumably records all relevant terrain features, including objects, locations, relations, and other attributes. Most typically these are digital recodings of maps; i. e., abstractions from representations which are themselves abstractions. The intent with such data sets has often been to record maps in ways that support subsequent display, perhaps to simply reproduce the original map, or consistency checking to ascertain that all elements of the original map have been encoded. This has usually included recording gross feature categorizations and geometric positions from existing map manuscripts.

The other representation is the text describing a specific place, usually from the perspective of someone in the field, by reference to salient landmarks and the visible relations among them. This is based on the less abstract experience of the geography. Establishing correspondence between these representations requires matching patterns of objects and relations. What are the objects and relations that need to be matched, and how should they be represented?

This chapter compares the content of existing cartographic databases with the kind of data required for more human-oriented spatial reasoning. The primary question addressed here is whether current cartographic data bases, such as those distributed by the US Geological Survey, are appropriately rich in detail to facilitate automated comparison with textual descriptions. The approach is to by examine what theory in spatial cognition and practice in herbarium records suggest will be the important features and relations to represent, and compare these to the what is represented in available data sets.

Part of the answer lies in understanding how people think about geographic space. Literature on cognitive categories and linguistic representations of space are particularly relevant. Studies of cognitive categories (Rosch 1973, 1978; Lakoff 1987) reveal the non-Aristotelian nature of mental categories and support the notion of basic-level categories, offering a default level of abstraction for reasoning and discourse about objects and relations. Herskovits (1986) categorized the spatial relations represented by several common prepositions. Her "use types", proffered as characterizations of spatial relations upon which one might base computation, and her recognition that the archetypical meanings of locative phrases are "stretched" for use in any instance are very useful in the present context.

Finally, a style of representation that might be appropriate for interpreting location descriptions is discussed. Many authors (Minsky 1974, Havens and Mackworth 1983, McGranaghan 1985, and Shapiro and Rapaport 1987, for examples) have suggested representation schemes that could support the pattern matching requirement suggested here. Indications are that future cartographic data sets will have richer structures, intended to model geographic more than map reality.

2. Background

The context for this work has been described in detail elsewhere (McGranaghan and Wester 1988; McGranaghan 1989a, 1989b). In brief, a set of botanical specimens had been collected at points, the locations of which were recorded in textual descriptions. The descriptions neither included geographic coordinates, nor followed a specified grammar. We were asked to map these locations using standard computer cartographic software. This required that coordinates be derived for the collection site points. Lyndon Wester and Warren Wagner spent considerable effort in manually interpreting the descriptions and in the process we recognized that this process should be automated.

2. 1. LABEL CONTENTS

One thousand three hundred forty location descriptions were drawn from specimen labels in the Bishop Museum's Herbarium Pacificum. Each label contains an accession number, identification (genus, species, and sub-specific taxa), collector, preservation method, and a description of where the specimen was collected. This description is usually terse but typically includes: the name of a country (or island, as in Hawaii), a locality (the name of a physical feature or land division), the site's approximate elevation, and, on about 75% of the labels, more detailed locational information. Often it contains directions for reaching the site. Prepositions, proper names of features and places, generic feature names, metric relational terms, and non-metric relational terms figure heavily in such descriptions. Habitat information such as ground cover, soil type and moisture, is also often present. The labels do not contain site coordinates.

2.2. TWO APPROACHES

Two approaches to automation have been examined (McGranaghan 1989b). The first was a brute force name matching algorithm which relied on matching place names from labels with those in a digital gazetteer (the Geographic Names Information System described below). The second attempted to use grammatical structure in the descriptions to gain meaning beyond the bare place names. Each of these approaches has some merit but neither is completely satisfactory.

2.3. SOME LESSONS

There are a number of difficulties with these approaches. Obviously, if the gazetteer database does not contain names from the label, there is no way to tie the site to a position. The grammar-based approach might use more of the label's information, but the current parser does not differentiate between prepositions. As we will see shortly, identifying a particular preposition points to a family of relations rather than a specific spatial relation; another knowledge base is needed to recognize the relations prepositions encode.

Preposition use involves knowledge of spatial relations and objects. In everyday discourse there is considerable redundancy in using prepositions due to our expectations about the possible relations among the objects. In the examples below, it is easy to fill-in the appropriate leading preposition when we know what the features are:

_?_ Palolo Valley
_?_ Koolau Mountains
_?_ summit ridge
_?_ Nuuanu Pali
_?_ Waialae Iki

but more difficult when the objects are not familiar. Knowing that "Pali" means ,'very sharp ridge" makes the use of "on" - in the fourth example much more likely. The use of "at" in the last example (without knowing what "Iki" means) would indicate that Waialae Iki is generalizable to a point.

We have a sense that in discourse: 1) we communicate about things and the relations among them; 2) often, prepositions code the relations, but knowledge of objects also informs us of the possible spatial relations between things; and 3) there is a need for sufficiently expressive representations for objects and the relations among them. As we shall see, it is not clear that existing databases store the correct knowledge for supporting human-like reasoning about geographic objects and relations.

3. Stores of Geographic Knowledge

The typical location specifying content of an herbarium label was introduced in the previous section. In this section I consider two other stores of locational information. The section opens with a brief view of spatial cognition and mental maps. These are likely to be closely related to the herbarium label contents. Then the remainder of the section introduces digital cartographic data sets currently available from the USGS.

3. 1. MENTAL REPRESENTATIONS OF GEOGRAPHIC SPACE

There is considerable evidence that people think about geographic space in terms of the objects in it and the relations among them, rather than in terms of systematic coordinates (Lynch 1964; Downs and Stea 1977; Talmy 1983, 1987, 1989; Herskovits 1986). Some places have salience as points or areas and serve as landmarks, these are joined by paths and separated by edges. A mental map can be constructed as a network of these objects and relations. A position can be specified by naming a landmark and then adding offsets and constraints to the description. The construction and use of cognitive models of space based on this concept has proved useful in geography, psychology, cognitive science, Al and linguistics (see for examples: Kuipers 1978, Elliot and Lesk 1982, and Goldin and Thorndyke 1982). Whether or not peoples' mental representations of space are ultimately network-like views, there seems to be sufficient reason to include both objective geometric representations and cognitively salient topological relations (or the ability to compute one from the other) in the database against which locations will be matched.

Lakoff (1987) provides an excellent synthesis of recent work on mental categories. The arguments that individuals organize knowledge in structures he terms idealized cognitive models (ICM) rather than in structures like those of classical logic and set theory are compelling. ICMs are complexes of propositional structures, imageschematic structures, metaphoric mappings, and metonymic mappings (Lakoff 1987 p 68). They do not follow the rigid hierarchy of Aristotelian classes, but rather, allow fuzzy and overlapping categories. These structures plausibly support effects like Rosch's "basic level categories." In addition to being psychologically satisfying, ICMs are computationally tractable.

The forgoing indicates that people think about space in a ways that are at once concrete and abstract. The things that we think about, both the objects and the relations among them, tend to be given, and probably seem clearly evident in the environment. But they are not easily amenable to definition by ridged rules into cleanly distinct categories. To give an example, it is very easy to conceive of a "hill" feature or a "mountain" feature, but it is difficult to say exactly what would differentiate the two. How are these ambiguities handled in standard digital cartographic data sets?

3.2. DIGITAL CARTOGRAPHIC DATA BASES

The United States Geological Survey (USGS) has produced four data sets, Digital Line Graphs, Digital Elevation Models, Geographic Names Information System, and the Land Use and Land Cover database, which are distributed for the by the National Cartographic Information Center (NCIC). These are representative of the types of geocartographic databases available today. They record a combination of geometry, topology, classification, attributes, and feature names, intended for the most part to support mapping. They emphasize geometric relations, and exploit topological relations more commonly to check data consistency than to guide spatial search. In this paper we will be concerned with three of these data sets.

3.2.1 Digital Line Graphs.

The USGS Digital Line Graph (DLG) files are the the first high resolution digital cartographic database to cover the entire US. They record the planimetry and categorization of roads, transportation facilities, hydrography, cultural features and boundaries. The intent has been to encode the information from the maps. Features are conceptualized as points, lines or areas at the fixed scale of the map. Positions are given as geometric coordinates, and the topological relations of the planar graph of the map objects is recorded explicitly. An object, i.e. the abstract geometric entity, may have several uses or functions. For instance, the object may be both the center line of a stream and a boundary between municipalities. In general, each distinct map symbol (not each occurrence of the symbol) will have a unique code.

The DLG feature codes are hierarchical. A two-level hierarchy provides both coarse and very fine categorization. The major code divides features into the twenty broad categories listed in (USGS 1985). A major code of 030 indicates a stream while 100 indicates a road. Within each of these, minor codes distinguish finer categorical division. Within the 030 category, a minor code of 0201 indicates an intermittent stream and a minor code of 0203 indicates a perennial stream. These feature codes were revised in 1983

(USGS 1985), and may be changed again for DLG-E, an enhanced version of DLG (Guptil 1991), but they remain of interest because of current availability and the inertia that will instill.

3.2.2 . Digital Elevation Models.

Digital Elevation Models (DEM) record terrain configuration as surface elevations (spot heights) on a regular grid. Geometric position in three dimensions is explicit and the topological relations between adjacent points are implicit in the regular grid structure. They do not identify, name or label parts of the surface. They only record positions in three dimensions.

Digital Elevation Models (DEM) are typically used to calculate slope, height, aspect, and volume (as for cut & fill operations). They are also used to generate contours, but they represent the terrain in a relatively sterile, geometric form. There are no features per say in a DEM. That is, none are given explicitly in the data base, but there are a number of terrain features (peaks, ridges, mountains, draws, spurs, saddles, etc.) that could potentially be recognized in the elevation grid.

3.2.3. Geographic Names Information System.

The Geographic Names Information System (GNIS) relates a name, a feature class, latitude longitude coordinate pairs, and other data for places, features, and areas which are identified by a proper name on USGS 1:24,000 scale topographic maps. In later versions of the data, historical names are included.

GNIS place names and coordinates are clearly useful search keys, but the feature class codes are also valuable in abstract knowledge representations. There are sixty three feature classes (see USGS 1987). Each feature class has a name, a definition, and usually a list of terms that are either synonyms, or members of the named class. Examination reveals that the categories are not classically hierarchical. They are not mutually exclusive, not exhaustive of common terrain features, and not at the same level of categorical abstraction.

The categories differ widely in their number of members. Geyser, arch and oilfield are defined very narrowly and have neither synonyms nor class members while "summit" has 26 synonyms and/or class members (including mountain, table, and rock) and valley has eleven (including gulch, gulf, chasm, ravine, cove, and draw).

There are a number of cases in which GNIS categorical hierarchies seem to be distorted. The term "mountain" is in a class called "summit", even though it seems to be the basic-level term and "summit" a subordinate class. Similar situations are found in several other classes. The "beach" category includes "coast, "shore" and "strand". "Bar" includes "ledge", "reef", "sandbar", "shoal", and "spit". "Coulee", "draw", "gully" and "wash" are all members of "arroyo".

Categories overlap. Both the "valley" and the "bay" categories have "gulf" and "cove" as members. "Stream" and "gut" both include "creek". "Rock" is in both "island" and "summit". "Lake" is defined as a class of natural inland water bodies, but also is a member of the "reservoir" class, where it is defined to be manmade.

Several common geographic features are not included in the data set. These include: road or highway, desert, dune, drumlin, esker, kame, kettle, and morrain. The category "area" serves as a catch-all for "any of several areally extensive natural features not included in other categories" and includes "delta". "Dessert" is not included elsewhere so presumably would also fall into this category.

4. Representation of Terrain Features

4.1 MENTAL REPRESENTATIONS

The spatial cognition literature suggests the need to represent terrain features that are landmarks, paths and areas, as a minimum. Introspection (admittedly suspect) provides a relatively small set of basic-level terrain features. Perhaps: mountain, valley, ridge, stream, road, shore, ocean, lake, and flat would suffice. Each of these terms refers to a set of objects that admit of a range of differences, but are related by some conceptual structure.

4.2 HERBARIUM LABELS

The Herbarium Pacificum labels refer to a slightly different set of features. The most frequently mentioned of which are (with number of uses): trail (120), mountain (114), range (106), gulch (62), valley (56), ridge (54), stream (54), slope (47), forest (43), head (32), summit (28), fork (21), reserve (21), bottom (20), bank (18), pu (hill) (21), branch (9), cliff (8), pali (8), ravine (5), ditch (5), gully (3), gap (3), gorge (2), declivity (2), and road (2). Several terms (pool, border, flume, peak, tributary, and rim) were used only once in the sampled labels . These features, though nowhere defined in the labels - and only presumed to be understood, should be represented in any database to which one would like to register the descriptions. The following section considers how well represented these features are in standard digital cartographic data bases.

4.3. DIGITAL CARTOGRAPHIC DATA SETS

4.3. 1. Digital Line Graph Features.

The list of base categories in (USGS 1985) is reasonably close to a set of basic-level landscape elements except for lacking the surface configuration features (e.g., mountains, valleys and slopes).It makes cognitively satisfying distinctions, such as those among hydrographic features, in separating linear water bodies, typically non-linear ones, inland wetlands and coastal features. Similar comments apply to the transportation categories.

The minor codes make finer distinctions than are usually found in the herbarium labels. However, this level of detail could be used to aid reasoning about locations.

4.3.2. DEM Features

Representation of terrain features in a DEM depends on the interaction of the grid spacing and its placement with the feature's size and location. If a terrain feature is large enough that is sampled in the elevation grid of a DEM then it is represented. The problem is recognizing that it is there and associating a name with it. There is some progress in this area but, so far, results seem more suited for specific purposes than for general feature extraction. Chen and Guevara (1987), Haralick and Shapiro (1980), Mark and O'Callaghan (1984), and Riazanoff et al. (1988) have discussed extracting terrain features from digital elevation models based on configurations of elevations. Typically, these studies have dealt with course lines and ridges. Frank et al. (1986) describe attempts to formally define terrain features as patterns in elevation data.

The techniques required to find such patterns are computationally intensive, not involving simple item to item comparison (as with search keys) but rather requiring comparison of assemblages of configurations of values. They are made even less attractive by the difficulty in defining features in terms of such patterns. Defining “mountain" such that: 1) it can be recognized as a pattern of elevations, and 2) people would generally agree that objects selected by the criteria are in fact mountains, is not easy. Determining the spatial limits of a "valley" involves determining where one shifts from being "in the valley" to being "on the ridge". Still, because several terrain features are not represented in any of the other standard data sets but are derivable from DEM data, it is clear that these data and techniques must be included in interpreting locality descriptions.

4.3.3. GNIS Features.

The categories for terrain features used in GNIS are not as cleanly organized as those in DLG data. The configuration of the terrain is not recorded, but the names of a number of important classes of terrain elements are related to coordinates. This suggests that for symbolic processing, GNIS does provide useful information.

There is fair intersection between the types of features named in the herbarium labels and used as (or in) GNIS categories. Trail, valley, ridge, slope, stream, spring, summit, cliff, falls, woods, forest and reserve are all used in common, though the intended meanings may be different in the two uses. It is interesting to note that “slope" is often present in the labels and appears as a GNIS feature class but that it is used to classify only two objects in the GNIS data file for Hawaii.

5. Representation of Locative Relations

5. 1. MENTAL REPRESENTATIONS

The set of relations used to specify relative positions has been studied by a number of authors. Several studies have deduced minimal sets of terms purported to describe all of the spatial relations two objects might possibly share. Freeman (1975) suggested the following as a complete set of spatial relations for objects in an image:

left of
beside
below
in front of
far
between
outside

right of
above
behind
near
touching
inside

This list was modified for geographic space by Peuquet and Zhan (1987) to include the cardinal directions (north, south, east and west). The problems of defining these relations from coordinates has been discussed by Peuquet (1986, 1987). This approach to identifying a sufficient set of spatial relations has been discussed by Mark and Frank (1990).

Herskovits (1986) indicates that there are relatively few schema, or prototypical relations, to which a locative description might refer. The prototypical relation is extended ("stretched") for use in any instance, and reasoning based on the prototypical situation is tempered by context. She provides analyses of the relations 46 at," "in," and d4on," as well as of a set of projective relations that closely resembles that from Freeman (1975).

Use types (Herskovits 1986, p. 86 et seq.) are the conceptual relations that might be signaled by a preposition. In each of these, the position of some object of discourse is given relative to some other ground object. Herskovits clearly intended that the use types should capture the meanings of relations in a way that was amenable to machine representation and processing.

5. 1. 1. At.

The ideal meaning of "at" is for a point to coincide with another, Herskovits (1986, p. 128) identifies eight use types (see list below) which seem at plausible.

Use Types for "at":

Spatial entity at location
Spatial entity "at sea"
Spatial entity at generic place
Person at institution
Person using artifact
Spatial entity at landmark in highlighted medium
Physical object on line and indexically defined crosspath
Physical object at a distance from point, line, or plane

5.1.2 On.

Herskovits (1986, P. 140) gives "on" the ideal meaning:

for a geometrical construct X to be contiguous with a line or surface Y; if Y is the surface of an object Oy, and X is the space occupied by another object 0x for 0 y to support Ox.

and identifies these use types:

Spatial entity supported by physical object
Accident/object as part of physical object
Physical object attached to another
Physical object transported by a large vehicle
Physical object contiguous with another
Physical object contiguous with a wall
Physical object on part of itself
Physical object over another
Spatial entity located on geographical location
Physical or geometrical object contiguous with a line
Physical object contiguous with edge of a geographical area

5.1.3. In.

"In", denotes "inclusion of a geometric construct in a one-, two-, or threedimensional geometric construct" Herskovits (1986, p 149). Use types are:

- Spatial entity in container
- Gap/object "embedded" in physical object
- Physical object "in the air"
- Physical object in outline of another, or of a group of objects
- Spatial entity in part of space or environment
- Accident/object part of physical or geometric object
- Person in clothing
- Spatial entity in area
- Physical object in a roadway
- Person in institution
- Participant in institution

5.1.4. Projective Prepositions.

Herskovits' projective prepositions (Herskovits 1986, p. 156) capture the sense of how bodies might be related to each other in geographic space.

- Horizontal Plane
  - (at/on/to/by) the (left/right) of
  - (at/on/in/to/by) the (left/right) (hand) side of
  - (at/on/in/by/to) the (front/back/side) of
  - in (front/back) of
  - before/behind
  - (right/back/left) of

- Vertical Order
  - above/below
  - over/under
  - (at/on/in/by) the (top/bottom) of
  - on top of

5.2. HERBARIUM REPRESENTATIONS

This section examines the contents of the herbarium labels in light of the "use types" proposed by Herskovits (1986). Recognizing the use of implicit but tacit prepositions (see the discussion and examples above relating to the robustness of our discourse understanding), the following discussion treats explicit uses of these symbols. Do Herskovits' characterizations of "in," "on," " at, " and the projective relations correspond to the uses found on the herbarium labels?

5.2. 1. At.

The word "at" was used thirty-five times in the sampled labels. Several uses were not easily assigned to one of the use types and three of the use types ("entity at sea", "person at institution", and "person using artifact" were not used at all.

The most common use type was the "object at a distance (from a plane)" construct. Thirteen of the uses specified an elevation, as in "at an elevation of 1100-1500 ft.," or more succinctly, "at 500-700 feet." This use type is not altogether separable from the "object on line and indexically defined crosspath" uses such as, "Waikane Ditch Trail at 750-900 ft." It seems that some object knowledge is needed to separate them.

Several constructions were difficult to assign to use types. "At intake #24", refers to a landmark (but not in a highlighted medium) or to a location (but not specified in a coordinate system), signalling either the "spatial entity at landmark in highlighted medium" or the "spatial entity at location" construct.

"At head of Punaluu Valley", "at the edge of (trail/stream)", and "at the edge of [an area)" each involve both the "at generic place" and the "landmark in highlighted medium" constructs, but also seem to indicate projective relations which are not in Herskovits' set. The former, because head is generic and not really a landmark in the sense of being a named object in its own right, and the latter because "head" is an identifiable part of the medium (valley).

Examining it in terms of the projective prepositions seems reasonable (it is tacitly "at the head of ...") reveals that "head" in this construction is very similar to the "front/back/top/side" signifying object parts (Herskovits 1986, p. 175). Herskovits stipulated that object parts be defined geometrically, but in this construction, the object part is a functional or morphological unit.

5.2.2. 0n.

"On" was used 77 times in the herbarium sample. Herskovits includes two use types explicitly mentioning geographical locations and geographical areas, but surprisingly few of the herbarium's ostensibly geographic descriptions fit easily into one of these categories. Only one unambiguously specified "on a geographic location" ("Along Castle Trail, on Koolau Range"). Three of the use types, "object as part of physical object", "object transported by a large vehicle", and object on part of itself", were not used at all.

The most common use of "on" was to indicate an object (terrain feature) which literally supported the plant (and the collector), the "supporting object" schema. Most commonly (27 occurrences) this was "on [some kind of) slope". (There were two "on bank", uses that could be lumped with the "on slope" phrases if banks are taken to be a class of slopes.) This was so frequent as to lead one to suspect that "slope" is a basic-level category of terrain features. "On (some kind of) ridge", was used only seven times - surprising given the placement of trails and access in Hawaii. Five specimens came from "on the summit [or crest, or top]," where the feature was named in a separate phrase (for example, "Nuuanu, on the top") making the meaning very similar to a projective use type.

The next most common use was projective. "On (some kind of) side of [something)" describes 19 of the uses of "on". These were most often on the side of a valley, but about a third of the time were "contiguous with the edge of a geographical area." The "contiguous with a line" use type had only ten cases. There were four cases where "contiguous with a wall" constructs were used to indicate being against the (very steep) side or head of a valley.

5.2.3. In.

"In" was used 81 times in the data set. As with "on" and "at", several of the Herskovits use types are clearly represented and, others are not. The 'in a container' structure is the most common. Taking gulch (18), valley (6), gully (3), draw (1), ravine (1), crevice (2) and ditch (2) together as bowl-like containers, there were thirty-three uses in this category.

The "object in the outline of another" construct was also very common. There were 16 "in" (some kind of) woods, and 13 specimens "in" (some kind of) forest. Four of these being "in" rainforests, two "in" thickets, and one was "in [a] very deep dark forest". Often the community composition (kukui/guava/ohia) was included as a modifier. This construct would also seem to be the best placement of "in rocks", as the plants were almost certainly among rather than inside the rocks.

The "spatial entity in area" captures most of the remaining cases. Four cases were explicitly "in [some kind of) area", four were "in kipuka (remnant patches)", one "in Forest Reserve", one "in pasture", and three "in lava flows." "In shade" (two cases) also seems to fit this construct, although not easily.

One construction resonates between broadening the "object in roadway" construct to become an "object in corridor or conduit" construct, and forcing a stream to be considered as an areal, rather than a linear feature. "In stream" could perhaps be conceptualized as a plant as an obstacle in the path of the flowing water (see Herskovits 1986, p. 154). It also could be conceptualized as the stream being a container for the plant.

5.2.4. Projective Relations.

Somewhat surprisingly, there is a real bias for plants to come from the nether regions of areas. Ten specimens came from "back of (a valley, a farm, or a building) but none came from "front of" anything (though, as noted above, several were collected from "head of [usually a valley)"). "Before" and "behind" behave similarly. No plants were "before" anything, while three were from "behind (something).

Left and right are not used much in this sample. There was one "to right of [a sign post)". Most of the uses of "left" were due to one collector who used both "from left roadside" and "left of palm" in each of the labels for four specimens.

Projective use of cardinal directions was also somewhat less frequent than one might anticipate in such a geographical context. There were 9 "east of"s, 1 "west of", 7 "north of"s, and 1 "south of'. Even including the Hawaiian terms for directions in the local polar coordinate system does not capture many more uses; there was one “mauka of" and no uses of "ewa", "makai", or "Diamond Head" as directions.

The exercise of trying to use Herskovits' use types to categorize locative propositions on the herbarium labels was instructive. On reading through them, the use types seem plausible, and indeed almost all of the uses in this database can be fitted into one of the use types. But the exercise also showed that even in this restricted domain, where intent is closely circumscribed, there is ambiguity. Generally, the use types captured the sense I understood in the phrases but occasionally it was difficult to decide which one was being used. A description understanding system would either need some further specification for determining classes or would need to be able to deal with mutually overlapping categories.

5.3. DIGITAL CARTOGRAPHIC DATA SETS

None of the digital cartographic databases explicitly encodes rich inter-feature relations. Only the DLG data sets record inter-feature relations of any kind. There data contain some topological information from which adjacent regions and successive segments along a linear feature can be computed. Further, containment relations could be calculated for the objects which are defined as polygons in this data set.

DLG, DEM, and GNIS all use coordinates to locate features. From these it is possible to compute many distance and direction relationships. Unfortunately, establishing correspondence between these computed values and linguistic categories is not straightforward (Robinson and Wong 1987, Peuquet 1986, 1987, Peuquet and Zhan 1987). From the herbarium record, it seems that getting at cardinal directions, though easy, is not all that useful. The tolerances required to interpret "along," "at," "near" and the like are still mostly unknown. None of these data sets allow easy determination of "the ridges that surround Palolo Valley." Similarly, "which slopes are along Waiomao Stream" is not directly answerable. Yet, it seems that a geographic database should allow very easy answers to these questions.

6. A Different Approach to Geographic Databases

6. 1. FRAMES AND NETWORKS OF GEOGRAPHIC OBJECTS

Is there a representation that could make salient objects and relations directly accessible? Minsky is credited (Winston 1977, p 179 et seq) with the idea of frames, basic structures for representing stereotypical situations which are adapted to represent specific instances of those situations. Frames are sometimes thought of as collections of "slots" and the values which occupy them. The slots may be occupied by other particular values, sets of values, or even other frames. Facts, i.e. knowledge, can be represented in networks of frames.

Mackworth and Havens (1983) describe Mapsee2, a system designed to interpret visual scenes (geographical sketch maps) in terms of hierarchical networks of schemata (frames). This approach, they argue, avoids both representational and computational problems associated with network consistency based approaches. Each schema represents a class of objects which has generic properties and possible relations to other schemata. A scene interpretation is then built as a composition of simpler schemata. A complete representation of a scene might then be composed of (decomposed into) schemata for river systems, road systems and a mountain range. Each of these schemata is in turn composed of (decomposed into) other schemata. Instantiation of a schema can be driven from the bottom up, from the top down or in a hybrid mode (Havens and Mack-worth 1983, p 95).

Although they do not stress the point, in the spatial context, connectivity and other relations to other schema could be considered and used to build geographic regions. For instance a valley schema could be linked to surrounding ridge schemata and [possible] mouth schemata. Regions built in this way would be consistent with multiscale phenomena in mental map use, as in route planning/selection where space is chunked at various resolutions for different (parts of) reasoning processes.

McGranaghan (1985) suggested that a list-based data structure might be used to represent any geographic object. This would use slot-value pairs to represent an entity with an arbitrary set of attributes and connections to other entities.

```
(ENTITY name-id
        (TYPE (of_entity (modifier))
        (IN entity)
(ALONG linear_entity_identifier)
        (BEFORE entity)
        (PAST entity)
        (NEAR entity)
(CONTAINS list_of_component_entities)
(NEIGHBORS (entity direction distance)..( )..( ))
(ORIENTATION (directional_trend_of_entity)
(MIN-RECT list-of-min-bounding_rectangle_coords)
```

```
(COORDS list_of-x,y_coordinates)
(FUNCTIONS list_of_activities_at_entity)
(HAPPENED-THERE list_of_events)
(INTENT goal))
```

A more developed version of this idea has been presented in (Buisson 1989). There, an object-oriented database was created to support spatial reasoning in a GIS. The slot values could either be given explicitly or could be inferred as needed by pattern matching within the database or by computation.

This type of representation for geographic objects could be modified to include other or variable relations. One advantage of labeled slots over data in which position in a file defines how a value is interpreted (as in GNIS, DLG, and DEM formats) is that it is not necessary to anticipate all possible future slots when designing an object's representation.

6.2. SCHEMA AND OBJECT MATCHING

Given data representations which were structured more around the geographic than the map objects, the problem of finding the coordinates of a described place could be translated into the problem of matching patches of space represented as a net of geographic entities, their properties, and the relations among both. A geometrically constrained, propositionally structured representation might start as a set of all the meaningful terrain units in a region. These would be related by their adjacencies, by functional connections, and by metric relations. Nodes in this net might have terrain unit classifications, coordinates, names and other attributes attached. Each of these attributes would be connected into networks of relations among categories. The result might be thought of as a "hypermap" and should be able to represent a good deal of what consider the "meaning" of a place, beyond its coordinates.

Interpreting a description would -then be recast as finding where the structure of the description's representation matches a piece of the hypermap. Goodness of the match should be allowed to be imperfect. Either the description or the hypermap could have more or less detail than the other in a given case. Software for matching such representations is discussed in Buisson (1989).

7. Conclusions

Existing digital cartographic data sets represent map information, but they do not capture the meaning of the map symbols. Neither, of course, do the maps. Meaning is human. For the data sets to incorporate meaning, it must be present in the representation. Some of it is calculable (at some cost) from the data but other parts are not because human categories for reasoning about terrain are not now adequately defined, and may ultimately be undefinable. Geocartographic data must be made much richer.

People use terrain feature categories that are not easily discriminated by reference to the surface configuration portrayed in DEM or, for that matter in digital contour data. They note relationships among features that are not directly discerned from coordinates via analytic geometry. If computers are to perform tasks that are more like human processing of maps and terrain descriptions, they must be able to recognize and access features and relations that would be meaningful to people. It is not critical that the underlying data be exactly like human representations, only that the machine be able to transform between its internal representations and ones that are meaningful.

Adequate access to objects and relations is not available in the current data sets. Strategies for providing it should be examined. One approach would be to restructure the database(s), processing them into some representations that make salient objects and relations directly accessible. Another approach is to build a database management system that is capable of computing objects and relations from current data as needed. The restructuring option carries a heavy manual and computational cost at the outset and a larger storage requirement, but it seems to allow fine tuning of categories that then may remain stable. The DBMS option may allow smaller storage, but will require heavier computation and will be very dependent on having correct translations between representations. In either case the nature of human categories must be addressed and ranges of machine manipulable symbolic representations mapped to human meaning.

The problem of translating these cartographic data bases into fuller knowledge representations remains one of defining mental categories such that they can be adequately symbolized. The confusion in the categories in GNIS, for instance, reflects both cognitive style and the difficulty in interpreting speech acts. Symbols, i.e. words are used differently in different (linguistic and geographic) contexts. The apparently simple problem of naming terrain features, big, objectively given things in the environment, is not simple: ill-defined terms, and terms with multiple meanings make symbolic processing difficult.

Ultimately, digital databases will better reflect geographic reality. The objects and relations to be represented are well known. A finished data set would be a set of objects that contains every terrain feature that might be recognized and all of the relations among them that are likely to be needed. This clearly entails tighter integration of the information scattered in current data sets. Attaching proper names to features in DLG data and conceiving of the data as representing geographic rather than map objects are steps in the right direction which will be part of the DLG-E standard (Guptil 1991).

8. Acknowledgments

The author would like to thank the participants and organizers of the NCGIA Initiative-2 for the comments and discussion on some of the ideas presented in this paper, and acknowledge the support of NSF under NCGIA grant No. SES 88-10917 during the final preparation of the manuscript.

References

Bahrami, A., (1988), Designing Artificial Intelligence Based Software, Sigma Press, Wilmslow, England.

Buisson, L., (1989) Reasoning on Space with Object-Centered Knowledge Representations, in A. Buchman, 0. Gunther, T.R. Smith, and Y.F.Wang (editors) Design and Implementation of Large Spatial Databases, Springer-Verlag, New York.

Cercone, N., and G. McCalla, (1987), (editors) The Knowledge Frontier, SpringerVerlag, New York.

Chen, Z. and A. J. Guevara, (1987), "Systematic Selection of Very Important Points (VIP) from Digital Terrain Model for Constructing Triangular Irregular Networks", Proceedings of Auto-Carto 8, pp. 50-56.

Downs, R., and Stea, D., (1977) Maps in Minds: Reflections on Cognitive Mapping, Harper & Row, New York.

Egenhofer, M.J., and A.U. Frank, (1989) Object-oriented modeling in GIS: Inheritance and Propagation, Auto-Carto 9, Baltimore, MA, pp. 588-598.

Elliott, R. and Lesk, M. (1982) Route Finding in Street Maps by Computers and People. Proceedings, AAAI-82 Conference (American Association for Artificial Intelligence)

Frank, A., Palmer, B., and Robinson, V., (1986) Formal Methods for the Accurate Definition of Some Fundamental Terms in Physical Geography, Proceedings of the Second International Symposium on Spatial Data Handling, Seattle, WA. pp 583-599.

Freeman, J. (1975) The Modelling of Spatial Relations, Computer Graphics and Image Processing, v. 4, pp. 156-171.

Goldin, S., and Thorndyke, P., (1982) Simulating Navigation for Spatial Knowledge Acquisition. Human Factors , v. 24, pp. 457-471.

Guptil, S., (1991), (editor) An Enhanced Digital Line Graph, U.S. Geological Survey Circular 1048, Denver CO.

Grace, G., (1987), Linguistic Construction of Reality, Croom Helm, New York.

Haralick, R. and Shapiro, L., (1980), "A Spatial Data Structure", in H. Freeman and G. Pieroni (editors), Map Data Processing, Academic Press, New York.

Hausser, R., (1989), Computation of Language, Springer-Verlag, New York.

Havens, W., and A. Mackworth, (1983), "Representing Knowledge of the Visual World", Computer, v. 19, n. 10, pp. .

Herskovits, A., (1986), Language and Spatial Cognition, Cambridge University Press.

Kuipers, B., (1978) Modeling Spatial Knowledge. Cognitive Science v. 2, pp. 129-153.

Lakoff, G., (1987), Women, Fire, and Dangerous Things, University of Chicago Press.

Lindkvist, K., (1978), AT verses ON, IN, BY: on the Early History of Spatial AT, Almqvist & Wiksell International, Stockholm, Sweden.

Mark, D. and Frank, A., (1990) Concepts of Space and Spatial Language, in D. Maguire, D Rhind, and M. Goodchild, editors, Handbook of GIS, Longman, London.

McGranaghan, M., (1985), "Pattern, Process and a Geographic Language", Ontario Geography, n.25, pp. 15-27.

McGranaghan, M., and L. Wester, (1988), "Prototyping an Herbarium Collection Mapping System", Technical Papers: 1988 ACSM-ASPRS Annual Convention: GIS, v.5, pp. 232-238.

McGranaghan, M., (1989a), "Context-Free Recursive-Descent Parsing of Location Descriptive Text", Proceedings of Auto-Carto 9, Baltimore, MD, April 2-7, pp. 580-587.

McGranaghan, M., (1989b), "Incorporating Bio-Localities in a GIS", Proceedings of GISILIS '89, Orlando, FL., November 26-30, pp. 814-823.

Minsky, M., (1974), "A Framework for Representing Knowledge" in P. Winston (editor), The Psychology of Computer Vision, McGraw-Hill, New York.

Obermeier, K., (1989), Natural Language Processing Technologies in Artificial Intelligence, Ellis Horwood Ltd, Chichester, England.

O'Callaghan, J., and D. Mark, (1984), "The Extraction of Drainage Networks from Digital Elevation Data", Computer Vision, Graphics and Image Processing, v. 28, pp. 323-344.

Peuquet, D., (1986) The Use of Spatial Relationships to Aid Database Retrieval, Second International Symposium on Spatial Data Handling, Seattle WA. pp. 459-471.

Peuquet, D., (1988) Toward the Definition and Use of Complex Spatial Relationships, Third International Symposium on Spatial Data Handling, Sydney Australia. pp. 211-224.

Peuquet, D., and Zhan, C., (1987) "An Algorithm to Determine Direction Between Arbitrarily -Shaped Polygons in a Plane", Pattern Recognition, v. 20, pp. 65-74.

Riazanoff, S., B. Cerevelle, and J. Chorowicz, (1988), "Ridge and valley line extraction from digital elevation models", International Journal of Remote Sensing, v. 9, n. 6, pp. 1175-1183.

Robinson, R., and Wong, R., (1987), "Acquiring Approximate Representations of Spatial Relations: Theory and Results of Tests", Proceedings of Auto-Carto 8, pp. 604-622.

Rosch (Heider), E., (1973), "Natural Categories", Cognitive Psychology, v. 4, pp. 328350.

Rosch, E. (1978), Principles of Categorization in E. Rosch and B. B. Lloyd, editors,
Cognition and Categorization, Lawrence Erlbaum Associates, Hillsdale, NJ.

Shapiro, S., and W. Rapaport, (1987), "SNePS Considered as a Fully Intensional Propositional Semantic Network", pp 262-315, in Cercone and McCalla (eds),
The Knowledge Frontier.

Shapiro, S., and G. Woodmansee, (1971), "A Net Structure Based Relational QuestionAnswerer", Proceedings of the International Joint Conference on AI, Washington DC, pp 325-346.

Svorou, S., (1988), "Lexical Sources of Spatial Terms", Cognitive and Linguistic Aspects of Geographic Space: Report on a Workshop, compiled and edited by D. Mark, NCGIA, University of California, Santa Barbara.

Talmy, L (1983) How Language Structures Space, in H. Pick and L. Acredolo, editors,
Spatial Orientation: Theory, Research and Application.

USGS (1985), USGS Digital Cartographic Data Standards: Digital Line Graphs from 1:24,000-Scale Maps, USGS Survey circular 895-C, USGS Alexandria, VA.

USGS, (1987), Geographic Names Information System, (Data Users Guide 6), US Department of the Interior, Reston, VA.

Winograd, T., and F. Flores, (1986), Understanding Computers and Cognition, AddisonWesley, Reading, Mass.

Winston, P., (1977), Artificial Intelligence, Addison-Wesley.