Iowa Research Online
The University of Iowa's Institutional Repository

Transportation Policy Research

1-1-1997

Spatial Data Systems for Transportation Planning

Marc P. Armstrong
University of Iowa

Gerard Rushton
University of Iowa

Jayajit Chakraborty
University of Iowa

Please see article for additional authors.

Copyright © 1997 the authors

Hosted by Iowa Research Online. For more information please contact: lib-ir@uiowa.edu.

Spatial Data Systems for Transportation Planning

Spatial Data Systems for Transportation Planning

Marc P. Armstrong
Associate Professor
Department of Geography and
Program in Applied Mathematical and Computational Sciences

Gerard Rushton
Professor
Department of Geography

Jayajit Chakraborty
Research Assistant
Department of Geography

Allen Wayne Ibaugh
Research Assistant
Department of Geography

Amy J. Ruggles
Research Assistant
Department of Geography

Prepared by the
Public Policy Center

University of Iowa
1997

This study was funded by the University Transportation Centers Program of the U.S. Department of Transportation and the Iowa Department of Transportation. The conclusions are the independent products of university research and do not necessarily reflect the views of the funding agencies.

PREFACE

Transportation is an inherently spatial phenomenon. With recent advances in geographic information systems (GIS), spatial referencing of a host of transportation system attributes, performance characteristics, and usage patterns has become increasingly feasible. Geographic information systems enable such spatial data to be correlated, thereby facilitating a host of analyses in support of the transportation planning process. The objective of this research has been to develop a methodology for transforming and dynamically segmenting data. Dynamic segmentation enables transportation system attributes and associated data to be stored in separate tables and merged when a specific query requires a particular set of data to be considered. A major benefit of dynamic segmentation is that individual tables can be more easily updated when attributes, performance characteristics, or usage patterns change over time.

Applications of a progressive geographic database referencing system in transportation planning are vast. Summaries of system condition and performance can be made, and analyses of specific portions of a road system are facilitated.

Research for this project was carried out in the University of Iowa Department of Geography with assistance from the University's Public Policy Center. Funding was provided by the University Transportation Centers Program of the U.S. Department of Transportation, with supplemental funding contributed by the Iowa Department of Transportation.

ACKNOWLEDGMENTS

In the preface we mention financial support of our research by the U.S. Department of Transportation's University Transportation Centers Program and the Iowa Department of Transportation. These agencies have our gratitude for their support.

Research for this project was carried out in the University of Iowa Department of Geography with assistance from the University's Public Policy Center. We wish to acknowledge support provided by both entities. In particular, David Forkenbrock, Director of the Public Policy Center, provided advice and guidance during several stages of this project and Anita Makuluni provided indispensable assistance during the preparation of this report.

Michael Slagel also worked with us during the initial stages of the project. We thank him for his valuable assistance.

CONTENTS

PREFACE....iii

ACKNOWLEDGMENTS v

FIGURES ix

TABLES.... xi

CHAPTER 1: INTRODUCTION.... 1

CHAPTER 2: CHARACTERISTICS OF A SPATIAL DATA INFRASTRUCTURE FOR TRANSPORTATION PLANNING 3
- Background 4
- Methodology.... 6
- A typology of transportation queries.... 9
- Using enhanced TIGER files to support decision-making.... 12
- Summary.... 13

CHAPTER 3: METHODS USED TO DEVELOP THE SPATIAL DATA SYSTEM15
- Conversion of the Iowa DOT's primary road attribute data 16
- Linking the Iowa DOT base road file to TIGER files in a GIS using dynamic segmentation 22
- Alternative sources of backdrops for TIGER file updates 37

CHAPTER 4: TOWARD A CONCEPTUAL FRAMEWORK FOR THE VISUALIZATION OF NETWORK INFORMATION 41
- Network map types.... 42
- Elements of the framework.... 45
- Conclusion 52

REFERENCES 55

FIGURES

2–1. Study area 4

2–2. Dynamic segmentation methodology 9

2–3. Basic queries 10

2–4. Compound queries—concatenation 11

2–5. Analytical queries—spatial buffers 11

2–6. Analytical queries—network algorithmic 12

4–1. Stylized flow map 43

4–2. Attribute-driven base-map transformation to alleviate overlap of line symbols in congested areas 43

4–3. Conventionally projected network map 44

4–4. Advertising map providing directions to a specific location 44

4–5. Geographic and topological base-map transformations 49

TABLES

2–1. Study data sources 4

3–1. Atlas database structure 24

3–2. Column formulas for the topological checking procedure 28

3–3. Sample output of the topological checking procedure using Microsoft Excel 29

4–1. Classes and sub-types of network elements 47

CHAPTER 1

INTRODUCTION

For many years, the Iowa Department of Transportation (Iowa DOT) has maintained base record files (BRFs) containing attribute data pertaining to primary roads in Iowa. These records are composed of 178 attribute fields that contain information essential for maintenance planning, pavement management, policy analysis, and many other aspects of transportation planning. The geographic referencing of this database is of highway segments defined by their mileposts. Many new applications, including efficient management of the database itself, would be possible if it were made available as a fully geographically referenced dataset in which geometric representation were fully integrated with attribute information. The goal of this project was to address this problem by developing a methodology for associating the base record fields with their spatial location through the use of a geographic information system (GIS). A five-county corridor in eastern Iowa was selected for this purpose.

Data transformation methods and dynamic segmentation methods were used to link the base record files to a linear referencing system. The data fields in the base files were converted to a tabular format to enhance their compatibility with topographically integrated geographic encoding and referencing (TIGER) line files from the U.S. Bureau of the Census. The TIGER line files were updated using Highway and Transportation maps from the Iowa DOT as a reference. A spatial data editing procedure was implemented to improve the spatial accuracy of the TIGER files. Once this task was completed, linear feature data could be managed through the process of dynamic segmentation.

Digital street centerline files with nationwide coverage available from the U.S. Bureau of the Census have been enhanced geometrically to improve their geographic accuracy and have also been supplemented with numerous attributes to support the implementation of transportation-related spatial queries and analyses conducted in a spatial decision support system. The geometrical improvements were conducted in two ways: using digital orthophotographs as backdrops, and using digital files obtained from county engineers. The attribute data were created from thematically-oriented subsets of BRFs from the Iowa DOT, then related to the digital map through a process called dynamic segmentation. Through this process, thematic information is interpolated to the links of the digital road network, rendering both types of information more useful. To support ad hoc queries needed in decision-support contexts, we developed a theoretical framework of network-based queries.

Dynamic segmentation is a method for dealing with changing attribute data that occurs along a linear feature. Further, it is a method of partitioning lines or arcs contained within a geographic database referencing system to reflect their underlying attributes. In a dynamic segmentation system, attributes are stored in separate tables and merged with their geographic representation only when specific queries are performed. This method reduces data duplication because

additional segments are not added to the database every time attribute data changes.

Chapter 2 discusses the background and motivation for this research and summarizes its accomplishments. Chapter 3 provides details of the methods used to integrate the different information sources and develop a consistent geographic referencing of them. In Chapter 4 we show how new visualization tools can increase the utility of GIS-based transportation data systems. Finally, Chapter 5 presents brief conclusions that note the implications of this work in a wide variety of application areas.

CHAPTER 2

CHARACTERISTICS OF A SPATIAL DATA INFRASTRUCTURE FOR TRANSPORTATION PLANNING

Federal, state, and local agencies that are responsible for the nation's transportation infrastructure would benefit from an information system to assist them in managing this infrastructure. These agencies require access to information that is used to address a broad range of transportation-related questions. Geographically referenced transportation questions can be answered by an information system that combines spatial information, including road geometry and intersection topology, with information that describes the characteristics of the transportation system. Several generic types of queries must be supported by such a system. The first query type is designed to find all features in a database that have a specific set of user-defined characteristics. For example, users may wish to visualize the distribution of all road segments in a county that were resurfaced during the past three years. The second generic type of query is site-specific: planners may want to know a specific set of road network characteristics that are present at a given location or in a given area (e.g., the type of signage that is present at an intersection). In addition to these basic query types, users also may wish to formulate questions that require either the concatenation of a set of these simple queries, or the use of mathematical models to calculate solutions (e.g., spatial variations in the relationship between the volume of truck traffic and the condition of road surfaces in a county or multi-county area). Digital spatial databases also support the calculation of accurate distances between places, which is crucial to the development of accurate spatial interaction models used in transportation planning. Finally, these databases support the display of interactions among various socioeconomic and geo-demographic variables.

Several independent data products now exist for most areas that, together, provide the information needed to support this broad range of transportation-related queries. These data types serve as key elements in a digital spatial data infrastructure that is essential to informed transportation system maintenance and planning. TIGER files were developed to support data collection activities of the U.S. Bureau of the Census for the decennial census of population and housing. TIGER files contain geometrical, topological and address-range information, along with well-defined links to census attribute data. These are key characteristics in any transportation information system. When available, other data sources, including state and locally maintained road management information systems, and digital orthophoto products also contain a wealth of additional, up-to-date geometrical and attribute information. These highly complementary data sources need to be integrated into a single geographic information system (GIS) database, however, to support a broad range of queries. The purpose of this chapter is to describe the problems encountered in this process of data integration, to demonstrate how these problems can be

solved by constructing a prototype information system for a multi-county region, and to develop a typology of the kinds of spatial queries that can be implemented with such an integrated transportation information system.

BACKGROUND

The study area for this research consists of five contiguous counties (Cedar, Iowa, Johnson, Linn, and Scott) in eastern Iowa (Figure 2–1). These counties form a corridor that contains a mix of urban arterial and rural networks along with sections of interstate highway.

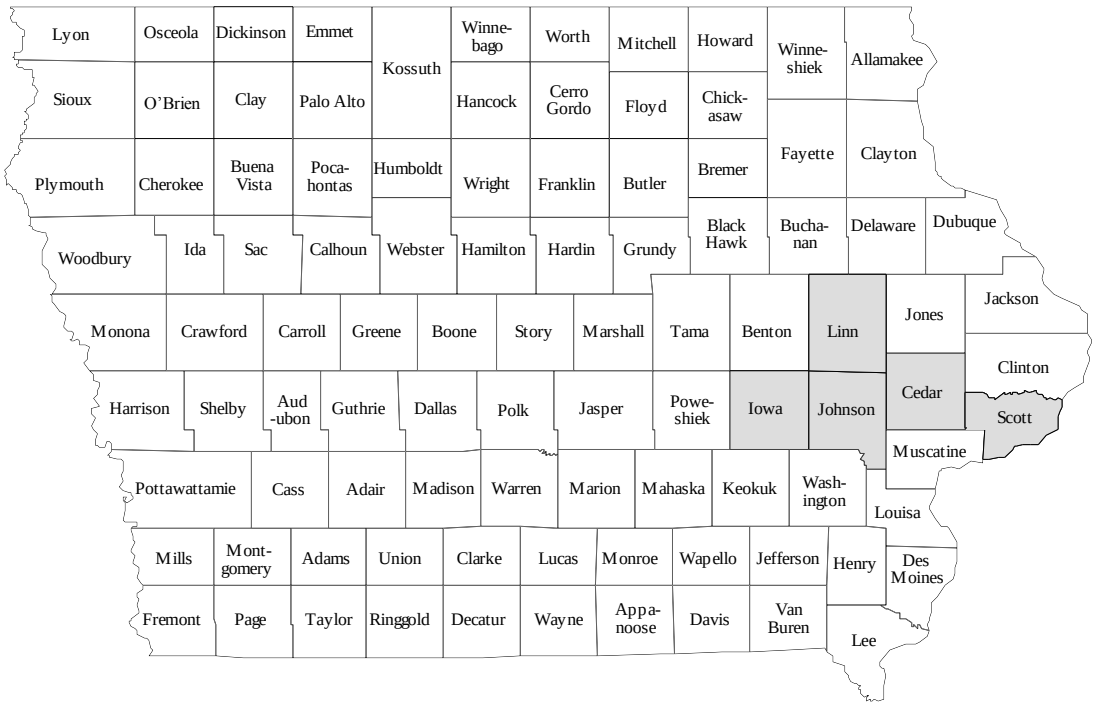

Figure 2–1. Study area

Data that describes the geometry, topology and attributes of the transportation network in this area are available from several sources (Table 2–1). Though TIGER files were used as the primary method of spatial referencing (U.S. Bureau of the Census 1993), a variety of other data sources were also used. County road maps were obtained from local engineering offices and scanned for input into the database. In addition, a digital orthophoto image for one part of the corridor was obtained. These raster-format data served as "digital backdrops" to the TIGER files and supported our update efforts by enabling us to check for errors in completeness and topology. In addition to these geographic data, the Iowa DOT maintains a comprehensive database, the BRF, that contains information on 178 attributes of the primary roads in this area.

Table 2–1. Study data sources

Data	Source	Key Characteristic(s)

TIGER files	U.S. Bureau of the Census	Topology; address ranges
Road management base file (BRF)	Iowa Department of Transportation	Attributes of road segments by milepost
Scanned orthophoto	Johnson County, Iowa	Geometrical accuracy and consistency
Digital county road map	Iowa Department of Transportation	Attributes of county highways

The locational characteristics of a large proportion of U.S. roads are now encoded through the efforts of the U.S. Bureau of the Census and the U.S. Geological Survey (USGS) as TIGER files, now available in CD-ROM format. In many areas, TIGER file geometry was derived from dual independent map encoding (DIME) files from the U.S. Bureau of the Census. These DIME files were developed to support the collection of information for the 1970 census of population (Holtzheimer 1983), which provided needed links between census attributes and geography. These files are now out-of-date. Moreover, they never possessed high levels of geometrical accuracy. In fact, the geometry present in DIME files was often an extreme topological abstraction of the existing road network; consequently, the geometry of a curved street segment might be represented only by the coordinates of its bounding intersections. These abstracted files were updated to support 1980 and 1990 decennial census activities using USGS digital line graphs (DLGs) and other sources. An attempt was made to add geometrical detail by introducing the concept of shape points that describe the form of road segments between intersection nodes (Broome and Meixler 1990).

The use of TIGER files to develop the spatial data infrastructure to support transportation planning is advantageous to practitioners and researchers in several ways. First, the files were developed to support Census Bureau data collection activities; these files are therefore not copyrighted and can be copied freely. Second, data for the entire nation is available. Third, the files are widely available and many GIS software programs are designed to use them. Fourth, TIGER files contain a wealth of existing attribute information such as address ranges for many streets. These attributes permit a wide range of non-geographically referenced administrative records to be geo-referenced and used in planning applications. Fifth, TIGER files provide direct links to all levels of census geography, and facilitate the use of census socioeconomic information. Finally, the topological structure of TIGER files supports the process of performing data integrity checks (e.g., ensuring network connectivity at road intersections).

Despite these clear advantages, TIGER files present pervasive problems when used in unmodified form (Rudnicki 1992). First, information in the original files is often incomplete; many roads may have been constructed between the time the TIGER files were compiled and their release. In locales without formal programs to ensure the regular maintenance of TIGER files, many of the files are now out-of-date. In many urban areas, for example, large numbers of new subdivisions are missing from the files. This is a particularly vexing problem in terms of planning because a thorough assessment of the current state of the transportation system is not even available. Fortunately, the problem can be overcome by purchasing updated geometry from geographic data vendors (e.g., Lew and Klosterman 1992), or by updating the files. A second problem, one that varies in severity, is that the geometrical fidelity of the road network is quite low in places. The extreme geometrical abstraction found in TIGER files is often a

legacy of past (DIME) versions of the file. A third problem is that some of the road segment characteristics that are central to transportation applications (e.g., speed limits) are not normally recorded in the TIGER data files. To meet this need, ancillary information must be integrated into the TIGER framework. In this case, the Iowa DOT BRF file is used to illustrate this process. Finally, a suitable spatial editing method has not yet been developed for merging separate TIGER county files and efficiently editing out those parts of the road network that are not required for particular transportation planning purposes.

Strategies must be developed to overcome these limitations so that TIGER files can be used more effectively for transportation planning. This requires the definition of a set of critical road parameters that must be associated with each road segment. A procedure for updating the geometry of TIGER files so that they can be used in places that have experienced growth also must be developed. Integrating TIGER files with other data sources to build a more comprehensive spatial data infrastructure will provide transportation planners with needed baseline information. This will partly recover the substantial initial investment in the existing digital spatial data infrastructure and extend its capabilities.

METHODOLOGY

TIGER files, because of their complete areal coverage, can be used to provide a comprehensive “base geometry” for any region in the United States. To overcome problems when used in transportation analysis, we updated the TIGER files in our study area to conform to several sources of more accurate data. Our approach consists of several steps:

- **Obtain TIGER files for the study area.** We used a county-level tiling, then imported these TIGER files into a GIS software environment (TransCAD™) to build a base data layer.
- **Assess completeness of the files.** We were specifically interested in locating missing road segments. We were also interested in determining the geometrical characteristics of the TIGER data in the study area, the set of attributes present, and the completeness of the data.
- **Correct the TIGER file geometry using complementary data sources.**
- **Augment attribute information.** For this study, we used the Iowa DOT’s BRF database. Other information, such as speed limits, road width, and surface material, could be obtained from county highway departments and added to the TIGER framework.

We consider the second, third and fourth steps of this process in more detail in the following sections.

Geometrical and topological correction

An enhanced TIGER file must reflect the current geometry of the road network. To meet this goal, however, ancillary data sources are required. Though USGS digital orthophoto quadrangles are a preferred source for this ancillary data, they were not yet available in our study area. As they become available, they will provide a valuable resource for updating the transportation component of the spatial data infrastructure. Similar sources of higher spatial accuracy can be obtained from scanned state DOT maps or from other digital orthophoto sources.

These ancillary coverages enable users to identify roads that are either out of alignment in TIGER files or missing from them. In addition, when making geometrical connections, the topological structure of the network must be corrected because in some instances TIGER files show connectivity where none exists and vice versa.

Corrections can be made in several ways. We first investigated the difference between the original TIGER file and digital orthophotos obtained for Johnson County, Iowa, from the Johnson County Assessor's office. These planimetrically accurate orthophotos provided a "digital backdrop" against which updates were made to the TIGER files. Because the orthophotos reflect a more current state of the road network, it is possible to use them to add missing road segments and new roads to the TIGER file. As implemented, the orthophotos are input as a GIS layer that serves as a visual underlay to the TIGER files. The GIS software is then used to register the TIGER files to the orthophotos at well-defined locations (e.g., the center of intersections). After the registration step is completed, affine (e.g., rotate and scale) and projective (e.g., nonlinear rubber sheeting) geometric transformations are performed to bring TIGER files into conformance with the higher precision base. When geometrical connections are made, such as extending a street to meet at an intersection or correcting a road alignment, the topological structure of the network is corrected. We also obtained scanned digital maps produced by county highway departments for the counties in the study corridor. This provided us with an additional means for identifying nodes (intersections) and updating TIGER road segments and geometry.

Attribute information

A substantial quantity of information about the condition and use of transportation networks is collected by government agencies as a routine part of their operations. In some instances, this information is collected solely to meet a particular, and often limited, objective and little or no attention is devoted to obtaining an understanding about how these different information sources can be integrated with other data to improve the overall effectiveness of management and planning activities. A common frame of geographic reference, such as TIGER files, permits the integration of attribute information from different sources, thereby expanding the usefulness of the data. Note, however, that only a limited number of specialized GIS software programs support the specialized functions required to perform necessary network data handling operations.

In many cases, hundreds of attributes (e.g., type of road surface) are used to describe highway networks. Managing linear feature data becomes difficult when these attributes change at different locations along a defined segment. If small linear segments are established and each has a unique combination of attributes, and if for each of these segments all of their attributes are stored, many of these segments will have large numbers of identical attributes. This redundancy is excessive, wasteful, and can violate good database management practices. Also, relating all point and line features to a reference system and predefining sections for all possible combinations of all attributes is cumbersome. Consider, for example, Dueker and Vrana (1992), who note that linear data may be referenced by:

- Reference markers (road name and milepoint),
- Sections (short sections of highways),

- Routes (collections of sections),
- Control sections (uniquely identified collections of routes or sections), and
- Chains (topological links in a network, usually related to sections).

Because different thematic databases may each employ a unique referencing strategy, a database that supports a wide range of transportation planning analyses must be able to provide a means for linking and integrating data collected using each of these different strategies. Our methodology, developed to add attributes to the existing TIGER files, relies on the use of a widely accepted method that supports the process of database integration. This approach, known as dynamic segmentation (Dueker 1987; Nyerges 1990; Dueker and Vrana 1992), is used to link attribute data collected using alternative methods of georeferencing to the road network, and is thus able to support the establishment of linkages between attribute information and the base skeleton provided by the TIGER files.

Dynamic segmentation limits data redundancy because attributes are stored in separate tables and merged with a geographic representation only when specific queries are performed. It enables us to change attribute characteristics at locations along road segments specified in the BRF. By using the enhanced TIGER data files and the BRF records, the resulting TIGER line segments have enhanced attribute characteristics (Figure 2–2). The specific set of additional characteristics was selected to support transportation management and planning activities.

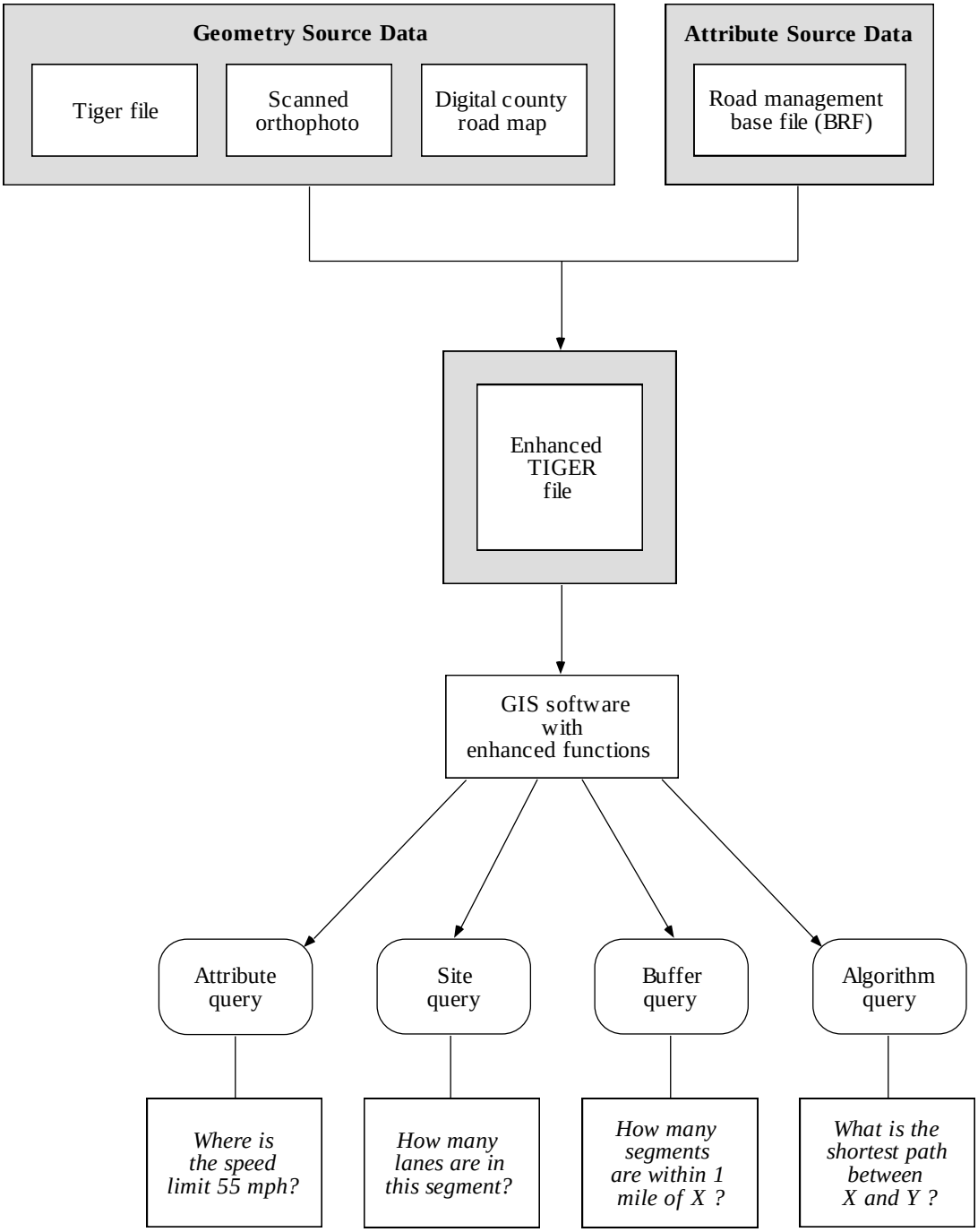

Figure 2–2. Dynamic segmentation methodology

A TYPOLOGY OF TRANSPORTATION QUERIES

In this section we develop a typology of transportation query types and illustrate their use. This typology is meant to reflect a basic collection of generic queries that are often required to support transportation planning. There are three broad categories of queries: basic, compound, and analytical.

Basic queries

Base attributes. This basic query type determines where one or more specific characteristics of a road system occurs on a network. The result is a table sorted by one of several variables (e.g., ID, route designation or county). This table is often visualized in map form with particular nodes or links highlighted in some way. For example, Figure 2–3 depicts the results from a show attribute query: *Show all roads resurfaced in the last ten years.* (In SQL form, the query would be formulated as: show X in Y where X is an attribute of the road system and Y is a record type.)

Site-specific. Here the objective is to determine combinations of characteristics that occur at a specific location. The location specified by the user may be determined by entering an ID (e.g., a road link ID) or may be specified interactively using a CRT display and an interactive pointing device such as a mouse. The location specified can be a point (node), line (link) or area. Figure 2–3 illustrates a site-specific query: *How many lanes does the selected road segment have?*

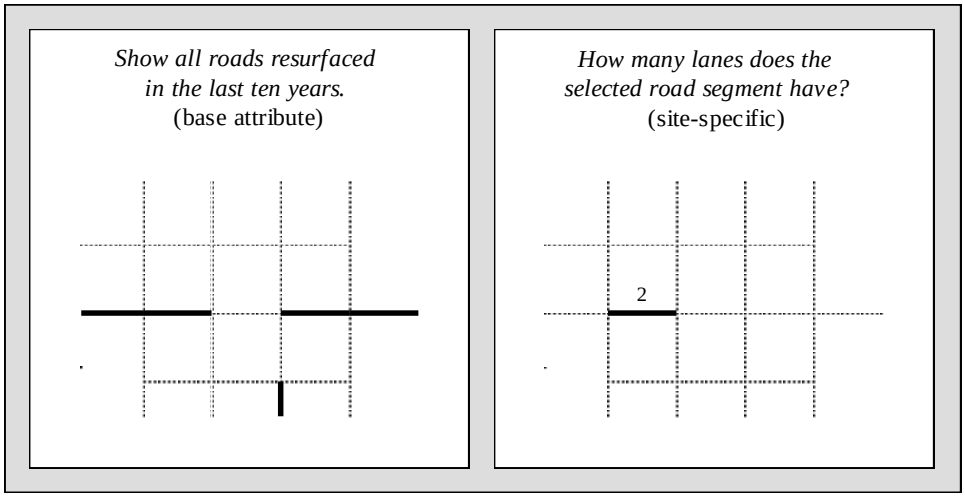

Figure 2–3. Basic queries

Compound queries

Concatenation. This compound query type is formed by logically concatenating two or more basic queries. The result is inferred, therefore, from two or more attributes. Figure 2–4, for example, illustrates how a concatenation query can be formed to determine the location of road segments where traffic volumes exceed the road design standards. Because multiple attributes are used in a variety of different combinations, the number of concatenation query types that can be generated from a database is normally quite large. A more complex concatenation query is illustrated by the query: *Which road segments are within 1.5 miles of school X with traffic volumes exceeding road design standards? (Figure 2–4).*

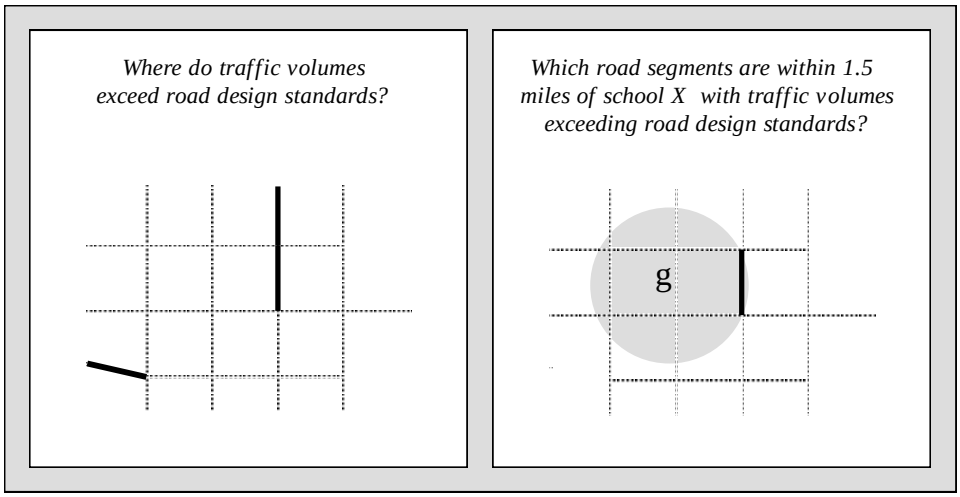

Figure 2–4. Compound queries—concatenation

Analytical queries

Spatial buffer. This query type uses the spatial analytical capabilities of a GIS to calculate an answer. The basic goal is to determine whether a location is within an area defined by the distance range of a point, line or area in the region. In this case a "seed location" is specified as a point, line or area, and a buffer zone is calculated. Point-in-polygon tests can then be used to determine whether specified locations are within the resulting buffer zone. Three types of spatial buffer queries are shown in Figure 2–5, which depicts the result of a query that would indicate the location of those schools that are within one mile of a primary highway. Note that in this case if any part of a link is within the buffer specified, the entire link is specified as being within the buffer region. In the second example, links are clipped to the extent of the buffer. The final example shows a linear buffer around a path in a network.

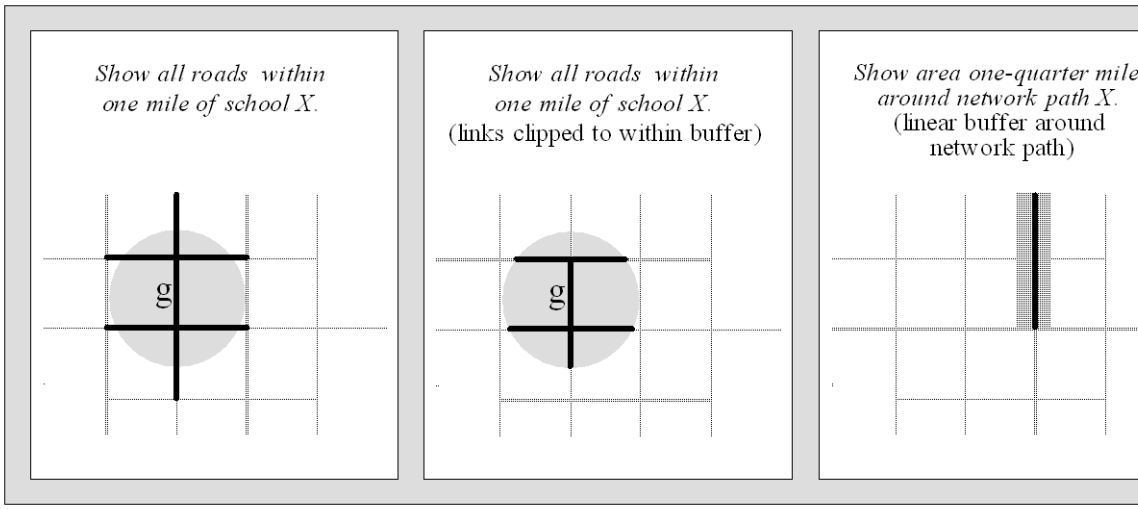

Figure 2–5. Analytical queries—spatial buffers

Network algorithmic. An algorithmic query uses the geometrical information contained in the GIS database to make analytical calculations that are part of an algorithmic solution process. The result may be reported in tabular form or as a map. The basic form of the query is: Where is [a condition determined by an implemented spatial algorithm]? In Figure 2–6, for example, the shortest route across the network between two points is shown. Note that in some cases this analytical query must have access to several types of transportation information (e.g., one-way streets and turning restrictions).

Network combinatorial algorithmic. A network combinatorial query is the most computationally demanding query type. In many instances, these queries cannot be accomplished in near-real-time because of the large number of combinations of nodes that must be examined to produce an optimal solution. Figure 2–6, for example, shows the optimal locations for fire stations in a network. Note that this query type also requires access to a variety of network information, is partially based on a network algorithmic query as a pre-processing step, and also requires the use of thematic data to determine the location and magnitude of demand.

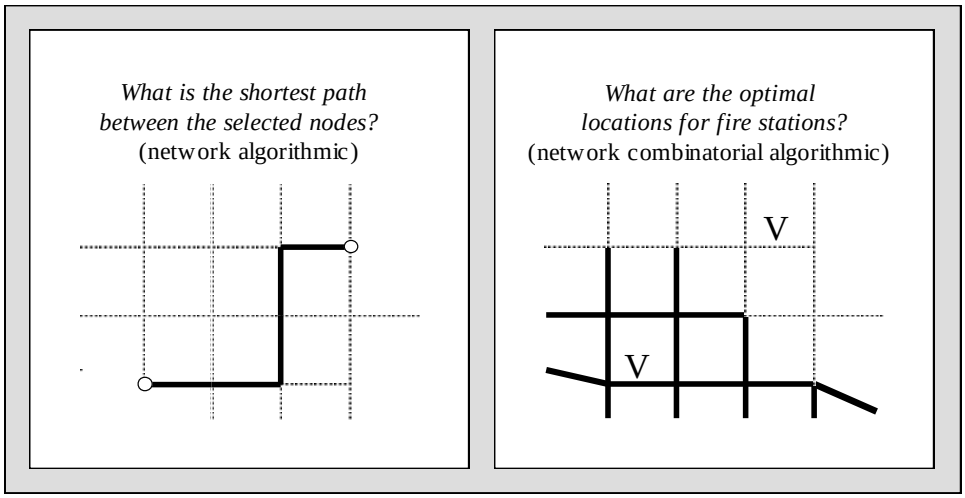

Figure 2–6. Analytical queries—network algorithmic

USING ENHANCED TIGER FILES TO SUPPORT DECISION-MAKING

Enhanced TIGER files can be used as the data kernel of a transportation decision support system (TDSS), which is typically built using a focused range of analytical queries linked to a spatial database, a graphics module and a user interface. The following steps were completed prior to our application tailoring efforts:

- the BRF flat file was imported into Paradox for re-formatting,
- control sections were created using section numbers and milepost data,
- the file was exported from Paradox as a comma-delimited ASCII file, and
- the ASCII file was imported into TransCAD™ for dynamic segmentation.

To demonstrate the flexibility of the use of the enhanced TIGER files in a TDSS context, we developed a set of prototypical applications: 1) a need study, 2) signage, 3) maintenance, 4) surfaces, and 5) traffic.

Each application consists of a different user view of the integrated database. Such views are built from logically linked fields that collectively serve to focus on a particular set of transportation planning tasks. In the case of the need study view, which is designed to support decisions about the adequacy of road segments, the following variables are included:

- route number,
- route sequence,
- sufficiency sequence,
- sufficiency typical, and
- curb and shoulder.

These fields contribute information relevant to need study tasks and related decisions that must be supported. In addition, derived fields also can be calculated and used in further analyses. In this case, the following fields were derived:

- foundation rating,
- surface rating, and
- drainage rating.

The surface rating, for example, indicates the wearing surface for a road section (obtained from the PSI, present servicability index, rating). It is evaluated from information that describes the condition of the pavement and the base courses and describes their strength, durability and rideability. Consideration is given to surface deterioration, failures, and excessive maintenance requirements. A four-tiered classification is used: excellent, good, fair, and poor.

These application views are used with the enhanced TIGER files to support several of the query types described earlier. For example, it is possible to formulate a set of attribute, site, buffer, and algorithm queries for the need study view; this would then serve as a TDSS with a specific focus on need studies. In such a TDSS, an attribute query might show the locations of those segments with a “poor” rating. A site query would show the rating at a given location along a road. A buffer query would find road maintenance garages that are within *n* miles of road segments with a “poor” classification. And an algorithm query would find a shortest path between a segment rated “poor” and a maintenance facility.

SUMMARY

We have developed a set of procedures that can be adopted by transportation planners to create a spatially integrated and dynamically segmented transportation database. In this approach, existing TIGER line databases are enhanced to improve their locational accuracy and topological integrity. The TIGER line segments are assigned an extended set of attributes that have been selected to support transportation planning activities. This results in an enhanced version of the geo-relational spatial data model in which each cartographic

feature (e.g., a road segment) has associated with it a collection of tabular data linked through an identifier.

The database can be used to create unique tables and maps that show relationships between data taken from different sources: it supports exploratory network data analysis. The procedures are flexible and can be readily adapted across a variety of hardware and software environments. The database can be extended further to incorporate other road network data referenced by mileposts and create a geo-referenced coverage that supports analysis and display. The TIGER database also provides address-matching capabilities that are able to support a wide variety of additional transportation planning activities.

CHAPTER 3

METHODS USED TO DEVELOP THE SPATIAL DATA SYSTEM

Iowa DOT base record data for primary roads in the five study counties (Cedar, Iowa, Johnson, Linn, and Scott) were linked to TIGER line segments through a method of data transformation and dynamic segmentation. In terms of data transformation, the Iowa DOT's base records were converted to a tabular format to enhance their compatibility with TIGER line files. Also, the TIGER line files were updated using the Iowa DOT's Highway and Transportation maps as a backdrop. A spatial data editing procedure was implemented to improve the spatial accuracy of the TIGER files. Once this was completed, management of the linear feature data was possible through the process of dynamic segmentation.

Dynamic segmentation is a method for dealing with changing attribute data that occurs along a linear feature. Further, it is a method of partitioning lines or arcs that are contained within a geographic database referencing system to reflect their underlying attributes. In a dynamic segmentation system, attributes are stored in separate tables and are only merged with their geographic representation when performing specific queries. In this way, additional segments are not required to be added to the database each time attribute data changes, therefore reducing data duplication.

Generally, dynamic segmentation addresses three data handling issues in geographic information systems. These issues include concepts of linkage, segmentation, and display/spatial analysis. Linkage refers to the joining of spatial and attribute data by relating records for linear spatial items and attribute data that is distance-referenced (Dueker and Vrana 1992). Segmentation creates new line or point items that may address different queries based on a choice of tables and values. Display and spatial analysis combines the two data handling methods above to yield an explicit referencing to the attribute-based linear items.

The common data management method for dynamic segmentation is a mileposting system. This is a type of linear referencing system where the geographic reference is a fixed point along a line feature. At this fixed point, information on signals, switches, junctions, and other transportation-related data are kept for future database queries. The geographic representation at each milepost location can be converted into a longitude and latitude coordinate. This can be called "geocoding by milepost." In the geocoding system, dynamic segmentation simply becomes a geocoding problem of interpolating along linear features using milepost ranges (Dueker et al. 1992). In the current study, dynamic segmentation was implemented to attach 178 fields of primary road attribute data to TIGER file primary roads in five Iowa counties.

The following tasks were completed to develop the spatial data infrastructure for transportation planning.

1) Define the five-county study corridor with full geographic coverage of the areas using TIGER line files version 4.0.
2) Convert the Iowa DOT's primary road attribute data from flat file format to tabular format to be imported into a transportation-related geographic analysis package.
3) Import and use Iowa DOT's highway and transportation maps as a backdrop to rectify and update TIGER line geometry.
4) Import and use scanned orthophoto quads to compare TIGER line files with orthophotography.
5) Create an Atlas database for each of the counties. This is a line database containing the complete geography of a road or utility network; it contains long segments with nodes located only at interchanges, at locations where route designations change, and at other important locations.
6) Label each segment in the study region with a route/control section number. (Route/control section numbers were developed by querying individual fields of the tabular formatted attribute data from the Iowa DOT);designate a link type for each primary road segment in the region (i.e., truncated line segments).
7) Prepare input milepost data and add values to the Atlas database using milepost tagging procedures. This procedure automatically assigns start milepost and end milepost values as well as CS/sequence numbers to segments in an Atlas database, based on input data from a comma-delimited text file.
8) Create milepost attribute tables with data describing the network characteristics at specific milepost locations.
9) Perform dynamic segmentation of the primary road segments in the study corridor to link the 178-field Iowa DOT primary road attribute data.

By implementing the above procedures, the study corridor primary road segments were successfully dynamically segmented to show various attribute data for each individual highway segment. The advantage of this transportation database is that it provides the minimum segmentation required to identify all homogeneous line segments, hence reducing data repetition. Also, the use of highly accurate background raster images for updating has resulted in edited and improved TIGER primary road segments for the study region.

CONVERSION OF THE IOWA DOT'S PRIMARY ROAD ATTRIBUTE DATA

This section contains a discussion on the methods employed to convert information from the Iowa DOT's base records (flat file format) to a tabular form for linkage with a transportation-related GIS package. This process can be divided broadly into two major parts:

1) converting the Iowa DOT ASCII files into a readable format and
2) manipulating the Iowa DOT data in Paradox 3.5.

Converting the Iowa DOT ASCII files into a readable format

The program FLIMPORT (Fixed Length Record Import, Release 3.0) was used to create a Paradox table from the fixed-length ASCII format records (Iowa DOT data). Based on information supplied in a separate specifications file, FLIMPORT creates a new table without the need for comma separation between fields.

Method of execution. FLIMPORT is an interactive program that enables one to select various options from Paradox-type menus. When FLIMPORT is invoked, the primary menu appears on the screen and provides the following options: create, modify, print, import, length, and exit. The *create* option is used to build the new import specification from scratch. Upon selecting this menu option, the user is prompted to enter the name of the specification file on which the desired task will operate. The name should conform to standard DOS file naming conventions and may contain a drive specifier, path name and/or file extension. Although any name can be used at this point, the maximum length is determined by the size of the highlighted entry field on the screen. This name should not, however, be confused with that of the Iowa DOT source (ASCII) file to be used.

Enter import specifications. After entering a file name for the specification, the computer displays a data entry screen divided into two parts. The upper half of the screen is for entering information about the table and file to be imported (the source file). The lower half is used for entering information about Paradox fields and the origin of the data used to fill them up.

1) **Table and source specifications.** This part of the screen contains four entry lines:

   - *Table name*. To begin, type into this field the name of the Paradox table that will contain records imported from the source file. The name should follow standard DOS file naming conventions but should not contain the *.DB* suffix, which is assumed by FLIMPORT. This table name will be used later in the Paradox application to view the Iowa DOT base records (the table name used for the project was *Tiger*).
   - *Action*. The next field offers three options for importing the source file. To build a new table (when no existing table has the name specified above), type “create” in the entry field.
   - *Source file name*. Into this field, enter the name of the file containing records to import (the Iowa DOT base records in this case). Drive specifiers, path names, and/or file extensions are allowed. For example, the source file name used for the project was c:\idot\idot.dat
   - *Record length*. Next, enter here the length of the records in the source file. As far as our project was concerned, all records were of the same length, and were specified as 514, equal to the total number of columns in the data file.

2) **Field specifications.** The lower part of the screen, or the *field spec* screen, contains four lines used to describe both Paradox and source record fields. Enter here the Paradox field name and type along with information about the location of the source data. There are four screen entry lines:

- *Field name*. Enter the name of the Paradox field that contains the imported data. The name must follow Paradox standards, so almost any character can be typed. For our project, there were a total of 178 fields. Typical examples of some of these field names are key year, system code, county no., state route/street no., surface type, and toll status.
- *Field type*. Valid field types are described in detail in the Paradox manual. The following is a brief summary.

  N = numeric format

  $ = dollar format

  S = integer format (e.g., key year or system code)

  Annn = character format of size nnn (e.g., toll status or municipal street name)
- *Starting position*. Indicate here the beginning position, or column, where the source data begins. FLIMPORT uses this number to find the data value to fill the Paradox field named above.
- *Field length*. Enter the length of the source field here, beginning at the *starting position* above. This tells FLIMPORT how many characters to gather for importing into *field name*.

Repeat this data entry process for all fields in the source file. After the specifications for all the fields (a total of 178 in this case) have been entered, invoke the specification menu (which is not visible when entering import features) by pressing function key <F10>. From the choices available, select the *Save* option to save the current specifications in a disk file whose name was entered earlier (see *method of execution*).

The import function. After having saved the specified table format, select the *Import* option from the primary menu to import the source file (c:\idot\idot.dat, in our case) as a Paradox table (whose name was specified earlier as *Tiger*). The screen will indicate when the import has been completed, and will note the presence of any errors.

Manipulation of Iowa DOT data in Paradox 3.5

To start Paradox 3.5, type the command “paradox” at the DOS prompt. The first menu displayed by Paradox is the main menu; other menus appear when selections are made from the main menu, or when function key <F10> is pressed. Pressing the <F10> key at any time will display the current menu, while pressing the <esc> key takes the user back to the previous menu.

The *View* option from the main menu is used to show tables on the screen. After the user chooses *View*, Paradox asks for the name of the table, and the user types in the table name used in FLIMPORT (i.e., *Tiger*) while entering import specifications for the Paradox table. What appears on the screen is actually an image of the table and any changes made to the image will not affect the table unless specified.

Add new fields. To aid the dynamic segmentation process, two new fields need to be added to the table named *Tiger* (Iowa DOT database) in Paradox.

1) **Control section.** The first new field was named *Control Section*, and was created by combining two fields already existing in the table, *System Code* and *State Route/Street No.* This requires the following steps:

- Select the *Ask* option on the main menu. This prompts the user for a table name (e.g., *Tiger* in our case) and the screen displays a query form for the table that contains all the fields but no data.
- Fill out a query form (type data into the fields). To select which fields will appear in the *answer* table (the results of the query are usually displayed in this temporary table), move to each field in the query form and press Checkmark (function key <F6>). To create a new field and retain all the existing ones, all the fields in the query form must be selected. To accomplish this, move the cursor to the leftmost field and press <F6> (the checkmark).
- Next, two mathematical calculations must be performed to combine the two fields, *system code* and *state route/street no.*

Step 1. First, multiply the *system code* field by the value 10,000 to create a number that can be used to form the *control section*. To accomplish this, type the following query in the *system code* field:

- press function key <F5> to signify an example query,
- type the example field name (i.e., “system code”), and
- type the expression “calc” followed by the example query name “system code” times (“x”) the constant value mentioned earlier (“10,000” in our example).

In final form this query in the system code field would look like this:

checkmark, <F5> system code, calc <F5> system code x 10000

After this line of information is typed in, press <F2> to process the query. The results will be written out to a temporary table named *answer*. Within this table, a new field will be created which Paradox automatically names *system code x 10000*. This table will also contain all the other fields if all fields have been checked by pressing <F6> with the cursor in the leftmost field.

Step 2. The second task in the formulation of the new *control section* field is to combine the two fields *system code x 10000* and *state route/street no.* To accomplish this, the user performs an *Ask* query in the newly created *answer* table. Again, press the <F6> key in the leftmost column to select all fields for the query. Next choose the *state route/street no.* field for the combination query by implementing the following statement in the same field:

checkmark, <F5>, state route/street no.

Perform a query by typing the following expression in the *system code x 10000* field:

checkmark, <F5> system code x 10000, calc state route/street no. + system code x 10000

Next press <F2> to process the query. The results will be written out to a new temporary table named *answer* (which overwrites the previous *answer* table). Within this table, a new field will be created which Paradox names *state route/street no. + system code x 10000*.

Step 3. The final step in the creation of the *control section* field is to restructure the table to eliminate unwanted fields (e.g., *system code x 10000*) and rename the newly created field "state route/street no. + system code x 10000." To accomplish this, first select the *Modify* function from the main menu, then select the subfunction *Restructure* inside *Modify*. When Paradox asks for the name of the table to be restructured, enter the name of the newly created *answer* table. Paradox also provides the option of either saving the table with the same name or a new one. It is important to rename the table using the *new* specification. Otherwise, each time a new query is conducted, an *answer* table will be created that will overwrite the table being restructured. Therefore assign a new name such as *Tiger1* to the *answer* table.

In the first phase of the restructuring process, the user presses the <page down> key to reach the redundant field (*system code x 10000*), which appears as the penultimate field in the table. To remove the field, line up the cursor with the field and press the <delete> button. The second phase of the process consists of renaming the *state route/street no. + system code x 10000* field to read "control section." Line the cursor up with the field (the last one in the table) and type the name "control section" in the name specifications. At this point, the restructuring process is complete; to process the query press <F2>. When Paradox asks whether to delete the penultimate field, press the <return> key to signify "yes." Finally, a new table is created (which we named *Tiger1*) that contains a new *control section* field.

2) **Milepost.** To create a second field name, multiply an already existing field in the table named *Beginning Milepoint* by a value of 1,000 in a process similar to the one for *system code x 10000*. To limit the redundancy of data storage, perform the query on the table named *Tiger1* (after going through the *restructure* functions, save the table as *Tiger1* and rename the new *beginning milepoint x 1000* field as *milepost* following the steps mentioned before.

At this point, the new table should contain two new fields (*control section* and *milepost*) that will be used later in the attribute matching process of Iowa DOT file manipulation.

Select specific records based on given field criteria. At some point in the data classification process it is important to select out various records based on a given criteria. If some of the records within a table are viewed as redundant or unreliable, they must be removed from the table using the Ask function of Paradox. Below are two examples specific to the Iowa DOT data file manipulation that deal specifically with such a problem.

1) **Eliminate records corresponding to ramps.** Data records corresponding to ramps on highway networks within the *Tiger1* table had to be eliminated

because ramps contributed inaccurate road lengths to the primary road segments (the main focus of the project). To remove these data records:

- Choose the *Ask* function from the main menu.
- Specify the table name *Tiger1*.
- Press the <F6> key in the leftmost field to checkmark all fields for the query.
- Type "< 49" in the *function code* field to eliminate the ramp data. This statement selects all records that have a function code less than 49 (all data records corresponding to ramps have a function code of 49 or greater).
- Press <F2> to activate the query. The results are written out to the temporary table called *answer*.

Once the query is complete, return to the main menu and rename the table by selecting the *Tools* function of the main menu. Choose the subfunction *Rename* and specify *answer* as the table to be renamed.

2) **Segregate records corresponding to county.** Up to this point the table *Tiger1* contains all the information for our study area (five counties in Iowa—Cedar, Iowa, Johnson, Linn, and Scott). To ease the data editing and manipulation process, the table *Tiger1* should be segregated into five different tables, based on county. The following steps result in the five desired tables:

   - As before, choose the *Ask* function from the main menu.
   - Specify the table *Tiger1*.
   - Press the <F6> function key in the leftmost field to checkmark all fields for the query.
   - Type "= 16" (for Cedar County) in the *county number* field to select out that specific county.

   Repeat the process outlined above for each remaining county. For our study area, we used the following expressions to produce a total of five new tables.

   = *48* (Iowa County)

   = *52* (Johnson County)

   = *57* (Linn County)

   = *82* (Scott County)

   After these specifications have been entered, press the <F2> key to process the query. The results are once again expressed in temporary tables called *answer*. These tables should be renamed to indicate their county names in the same fashion as mentioned before.

Sort the records based on specified criteria. Paradox allows the user to specify the order (ascending or descending) to sort records in a table. The following steps sort the individual county files based on the newly created fields *control section* and *milepost* in ascending order.

1) Choose the *Modify* function from the program's main menu.

2) Choose the *Sort* subfunction and specify the county file name.

3) Choose the *same* option to name the new sorted file as the county name.

4) Type the number “1” beside the field *control section* and number “2” next to the *milepost* field to sort the records in ascending order first by control section and then by milepost.

   NOTE: To sort the records in descending order, simply type the letter “D” after the numbers “1” and “2” (i.e., “1D” and “2D”).

After the specifications are complete, press the <F2> key to complete the sorting process. At this point, the new county files should be formatted and ready for implementation in the TIGER file dynamic segmentation process.

Convert the Paradox county tables to a comma-delimited ASCII file. To utilize the newly formatted county table data in a TIGER file within a GIS framework, it is necessary to convert the Paradox county tables to a comma-delimited ASCII file. The use of these files in a GIS database will be discussed in a later section. The process for converting files from Paradox is as follows.

1) Choose the *Tools* module from the program’s main menu.

2) Choose the *Export* subfunction of *Tools*.

3) Choose the *ASCII* function to export the paradox file to a *.txt* file.

4) Paradox will ask whether the file should be delimited. Choose *Delimited*.

5) Next, Paradox will display a prompt asking the user to enter the table name to be converted (e.g., *enter county name here:* Linn).

6) The final bit of information Paradox will require is a name for the converted file. Any name can be used. For this study we chose to use the same county names as those suggested by Paradox.

   At this point, all Paradox manipulation and data transfer procedures are complete and the user is ready to implement GIS applications of the converted files.

LINKING THE IOWA DOT BASE ROAD FILE TO TIGER FILES IN A GIS USING DYNAMIC SEGMENTATION

After a table for each county has been converted to a comma-delimited ASCII format, the stage is set for linking the database to the TIGER files in a GIS environment.

Managing linear feature data is difficult when values of different attributes change at different locations along the given road segments. Dynamic segmentation is considered to be a solution to the problem of managing linear data in a GIS, and deals with changing attribute data in an efficient manner that limits data redundancy. For our project, the GIS transportation package TransCAD (developed by Caliper Corporation) was used to link spatial and attribute data by relating records for linear spatial items and distance-referenced attribute data.

The six-step process for linking the Iowa DOT data and the rectified TIGER files for the five counties in TransCAD are summarized below.

1) Import TIGER files (from the CD-ROM) into TransCAD.

2) Create an Atlas database with appropriate geography for the road network.
3) Prepare a backdrop for updating TIGER files:
    - scan orthophotos and Iowa DOT base maps,
    - translate PCX images into TransCAD, and
    - register the image background.
4) Rectify the TIGER line files—both geometry and attribute information.
5) Complete topological checking and editing procedure.
6) Complete dynamic segmentation process:
    - Label each segment with a control section number and designation of link type using the on-screen edit function.
    - Prepare input milepost data and add values to the Atlas database using procedure MPTAG (input file).
    - Create milepost attribute tables (MATs) with data describing the network characteristics at specific milepost locations.
    - Group data fields associated with Atlas by creating text files listing field names and associated MAT file names.
    - Initiate TransCAD's dynamic segmentation procedure.

Importing TIGER files from the CD-ROM into TransCAD

1) Locate the directory that contains the county to be imported into TransCAD. Counties are listed by the federal information processing standard (FIPS) code. TIGER files on the CD-ROM are organized by state (Iowa's FIPS code is 19) and then the three-digit county code. For example, the complete path for Linn County's TIGER files (assuming the CD-ROM is in the *d:* drive) is *d:\19\113*. A complete listing of all FIPS codes can be found in a file on the CD-ROM of TIGER files. To view this file, type "EDIT FIPSCODE.ASC" at the *D:>* prompt.
2) Next, copy the TIGER files for the pertinent county to a directory on the *C:* drive. (Using our Linn County example, type "copy d:\19\113*.* c:\.")
3) Access the TransCAD directory and the TCBUILD utility by typing "tcbuild" at the TransCAD prompt (i.e., *c:\transcad>tcbuild*).
4) Select the *Translate* command from the menu board, followed by *Tiger/Line Set-Up*.
5) At the input screen (on the line that asks for TIGER FILE TYPE), indicate the type of TIGER file to be used (e.g., Precensus/census).
6) Next fill in the rest of the template indicating the FIPS code for the county being imported (e.g., *19113*), the names of the input TIGER/LINE files (e.g., *TGR19113.F41* and *TGR19113.F42* for the Linn County files that are of census file type), the drive and directory for both input and output files, and the type of translation to be performed (creating a database).
7) Press the <page down> key to specify the types of features to be included in the database (e.g., roads, power lines, or waterways).

8) Select the *Translate* command from the menu board, followed by *Tiger/LineCreate*. TransCAD will extract TIGER file information and write it to a format TransCAD can understand.

9) Once this is finished, any number of fields can be added before the actual building of the database. This is the time to add Atlas database fields.

Creation of an Atlas database with appropriate geography for the road network

The *Atlas Database* is a TransCAD line database containing complete geography for a road or utility network. The *Atlas* generally contains long segments with nodes located only at interchanges, where route designations change, and other important locations. The process involves the following steps.

1) Select *Edit* from the menu board, followed by *Layers* from the Edit menu.

2) Choose *Links* at the TransCAD prompt to select either *nodes* or *links*.

3) At the *Layer Information* screen, add five new fields to the existing number of fields:

   – Enter the number *5* at the *number of data fields* prompt.

   – Press <control> and <page down> to go to the last field.

   – Press <page up> five times to go to the first unnamed field (#9) and enter the appropriate Atlas database geography. The Atlas must contain five attribute fields (control section, type, start MP, end MP, and CS/sequence), with the names and formats indicated in Table 3–1.

Table 3–1. Atlas database structure

Field number	Name	Description
9	Control section	Long integer key field that contains the control section number (or route number) of the line segment
10	Type*	One-character field that designates the segment type (this designator distinguishes ramps, service roads, or rest areas that share control section numbers)
11	Start MP	Long integer field containing milepost value at beginning of segment
12	End MP	Long integer field containing milepost value at end of segment
13	CS/Sequence	Long integer key field containing a sequential number for each segment in a control section (this is the control section number linked with a sequence number)

*There is an existing field in the TIGER files called *Type*. Because we need to create a new field with the same name for dynamic segmentation, the existing field *Type* should be changed to something similar to *Segment Type*.

4) Select *Database* from the command menu board.

5) Select *Build* from the Database menu. TransCAD builds the database from the TIGER files with the new fields added.

Preparing a backdrop for updating TIGER files

Scan orthophotos and Iowa DOT base maps. Orthophotos were available from the County Auditor's office for only a small section of Johnson County. Each orthophoto represented an area of two miles by one mile. The highway and transportation maps (base maps) for each county were obtained from the state DOT.

Deskscan 2.0 for Windows was used to scan the orthophotos and maps. Each orthophoto was scanned breadthwise (a third at a time) so that it took three scans scan an entire orthophoto. The county base maps were reduced in size to fit an 8.5" by 11" sheet, and then scanned. Both orthophotos and base maps were scanned as PCX images and saved to floppy disks.

Translate PCX images into TransCAD. Any software package that can read PCX files can by used for this purpose. For the Iowa DOT project, Paintbrush on Windows was used, following these steps:

1) Copy the PCX images from floppy disks to the hard disk drive, and call up Paintbrush by double-clicking the appropriate icon.
2) Choose *File* from the menu board, then *Open* to bring up the PCX file that will be translated.
3) Select *Edit* from the menu board, followed by *Cursor Position* from the *Edit* menu.
4) With the *Cursor Position* on, select three easily-identifiable points on the map or PCX image (preferably highway intersections), and note their coordinates which appear as (column #, row #). Quit the application and return to the TransCAD prompt in DOS.
5) Access the TIGER file for the county that corresponds to the orthophoto or base map where three coordinates have been noted. Select *Query* from the menu board, then select *One or more*. Identify the three locations that correspond to the three points chosen previously in Paintbrush by pointing the arrow on the screen as close as possible to the original location of the points. Their coordinates (longitude, latitude) in TransCAD show up at the bottom of the screen, and should be noted. Next quit TransCAD and return to the DOS prompt.
6) Type "cd c:\transcad" to go to the TransCAD prompt. At the prompt, access the DECPCX utility (which translates PCX files) by typing "decpcx," the name of the PCX file to be translated, and a new name for the header file that TransCAD would use to access the image. For example:

   c:\transcad>decpcx picture1.pcx johnson

   where the PCX file to be translated is called *picture1*, and *johnson* is the name of the image header file to be created in TransCAD.
7) After pressing the <return> key, confirm that there are three known coordinates by typing "y" (yes). Next, enter the coordinates for each of three points that will come up in the following order:

scan line	row # from Paintbrush
pixel	column # from Paintbrush
lon (in millionths of degrees)	longitude from TransCAD

lat (in millionths of degrees) latitude from TransCAD

8) Once this is complete, access the TIGER file for the pertinent county to display the raster image backdrop. In TransCAD, choose *Display* from the menu board, followed by *Background*, and then *Image Background*. Select the appropriate image header file (named *johnson* in the example) from the pop-up menu to view the raster image of the orthophoto/base map, which appears as the background for the map display.

Register image background. TransCAD performs a mathematical transformation to register the raster and vector data. The registration process consists of the following steps.

1) Choose *Image Registration* (after selecting *Display* from the menu board, then *Background* as mentioned previously) to register the background image with the data in the TIGER database.
2) Choose three pairs of points. For each pair, select one point on the raster image and one point from the TIGER database to indicate the proper orientation and size of the image with respect to the data.
3) Once the image background has been redrawn using the new registration, the image registration can be saved permanently, using the *Save Image Settings* command from the *Background* menu.

To help speed up the display, the raster image is always displayed in the same orientation. The registration parameters are used to distort the display of vector data to yield the desired correspondence between raster and vector data.

Rectification of TIGER line files (both geometry and attribute information)

After the background image has been registered, the differences between the Iowa DOT background image and the TIGER line files can be viewed by zooming into areas of interest. But before doing that, one must select and highlight the primary roads in the TIGER files that will be examined.

1) Select *TransCAD* from the menu board, followed by *Conditions*.
2) Once in *Conditions*, set up four conditions by typing or choosing

Under condition:	Type or choose:
Condition name	the names of the primary roads to be examined
Data field	based on... *Name*
Operator	*Matches Exactly*
Value	the name of the highway, exactly as it appears in the TIGER database (e.g., Interstate 80 is *I–80*, and U.S. Highway 6 is *US Hwy 6*)

3) When all conditions have been set up, return to *Map Display* (TransCAD menu). TransCAD will automatically evaluate the conditions.
4) When *Map Display* appears, select *Several* and *On Condition*. From the choices available from the pop-up menu, choose *Union of Conditions*.

5) The new pop-up menu lists the names of the highways, all of which are selected by toggling. This highlights the necessary road segments on the map.

As mentioned previously, the differences between the Iowa DOT background image and the TIGER line files can be viewed by zooming into areas of interest. To zoom in TransCAD, use *Display*, then *Scale*, and finally *Zoom*.

Next, it is necessary to compare the TIGER file to the Iowa DOT background image and, where there are inconsistencies, edit the TIGER files on-line. The editing can involve various TransCAD functions such as Add, Join, and Move, depending on the discrepancy in the TIGER line files. For example, it may be necessary to move a TIGER line to match the background image properly. The following steps describe the process that would be followed.

1) Select *Projection* from the *Display* menu, and specify *State Plane* coordinates before making any editing changes.
2) Activate the *links* layer from the *Active Layers* prompt in the TransCAD menu.
3) Choose the *Move* function from the *Geography* menu.
4) Point to the line to move.
5) Press the <left> arrow button at the new starting node location.
6) Press the <right> arrow button any number of times to locate shape points.
7) Press the <left> arrow button at the new ending node location.

At this point, the line is moved. The nodes at either end of the line are moved to the new location and any other lines that connect to these nodes are adjusted accordingly. Other editing functions use similar procedures.

Topological checking and editing procedure using Microsoft Excel 4.0

During the course of the dynamic segmentation process, problems concerning the topological consistency of TIGER file data are likely to be encountered. TIGER files are notorious for their lack of completeness (e.g., missing line segments and/or non-connected nodes). For example, in Linn County, Iowa, several TIGER line segments were found that either were not connected at places they should be, or had several segments missing altogether. This causes enormous problems for the dynamic segmentation algorithm. When a route has a segment that is missing the next segment along that route or is not connected via a common node to the next segment in the route, the mileposting procedure (the algorithm responsible for assigning a milepost value to the route) ceases to function at the point where the connection is broken.

One method that was attempted was a visual “traversing” of the route to see if all of the segments had been mileposted correctly. This turned out to be very laborious and tedious. The TIGER file for Linn County had several segments that measured less that 0.02 miles (roughly 100 feet). Identifying and editing linear features at this scale on a county-wide basis is a time-consuming process.

In an attempt to fill the need for a topological checking procedure and in an attempt to devise a more automated inspection method, a simple topological checking and editing procedure was developed using Microsoft Excel 4.0. The procedure requires three inputs to work correctly. For each line segment to be

analyzed, the *To* and *From* nodes must be specified, along with the TIGER line segment ID number. The nodes are in the form of numeric identifiers. In Table 3–2, columns A and B contain the *To* and *From* nodes, while column C has the TIGER line segment ID number.

The functions used by the procedure to accomplish the topological check are found in columns D through G and in column M in Table 3–2. Simply put, columns D and E use a searching function that examines each node number in column A for a match in column B, and vice versa. If a match is found, which would indicate connectivity, a true response is returned. If no match is found, a non-available value message is returned.

If at least one non-match is found to exist for a line segment, the segment may not be topologically consistent, and the Boolean expression in column F is satisfied. Column F simply says, *Return value of true if column D or column E does not have a matching node*.

Table 3–2. Column formulas for the topological checking procedure

A	B	C	D	E	F	
1	3	84488171	=IF(MATCH(B1,$A$1:$A$404,0),1)	=IF(MATCH(A1,$B$1:$B$405,0),1)	=OR(ISNA(D1),ISNA(E1))	…
7	1	84488175	=IF(MATCH(B2,$A$1:$A$404,0),1)	=IF(MATCH(A2,$B$1:$B$405,0),1)	=OR(ISNA(D2),ISNA(E2))	…
8	7	84488178	=IF(MATCH(B3,$A$1:$A$404,0),1)	=IF(MATCH(A3,$B$1:$B$405,0),1)	=OR(ISNA(D3),ISNA(E3))	…
11	9	84488181	=IF(MATCH(B4,$A$1:$A$404,0),1)	=IF(MATCH(A4,$B$1:$B$405,0),1)	=OR(ISNA(D4),ISNA(E4))	…
3	11	84488182	=IF(MATCH(B5,$A$1:$A$404,0),1)	=IF(MATCH(A5,$B$1:$B$405,0),1)	=OR(ISNA(D5),ISNA(E5))	…
12	14	84488184	=IF(MATCH(B6,$A$1:$A$404,0),1)	=IF(MATCH(A6,$B$1:$B$405,0),1)	=OR(ISNA(D6),ISNA(E6))	…

	G	…	M
…	=IF(F1=TRUE,”The Following Segment ID is not topologically consistent: “,””)		=IF(F1=TRUE,C1,””)
…	=IF(F2=TRUE,”The Following Segment ID is not topologically consistent: “,””)		=IF(F2=TRUE,C2,””)
…	=IF(F3=TRUE,”The Following Segment ID is not topologically consistent: “,””)		=IF(F3=TRUE,C3,””)
…	=IF(F4=TRUE,”The Following Segment ID is not topologically consistent: “,””)		=IF(F4=TRUE,C4,””)
…	=IF(F5=TRUE,”The Following Segment ID is not topologically consistent: “,””)		=IF(F5=TRUE,C5,””)
…	=IF(F6=TRUE,”The Following Segment ID is not topologically consistent: “,””)		=IF(F6=TRUE,C6,””)

The procedure then moves to column G. Column G begins the process to identify segments that are not topologically consistent. Column G poses the question, *If column F is true, then print “The Following Segment ID is not topologically consistent: ”, else print nothing*. All of the segments that have at

least one non-matching node are identified at this point. Column M sets up an identification process for each segment. If column F is true, then column M returns the segment ID whose topological consistency is in question.

The analyst then prints columns G and M for a listing of all of the segments that may not be topologically consistent. Working from these segment ID numbers, the analyst can query the database to find these segments and examine the reason for their lack of topological consistency. Table 3–3 provides an example of output from a run of the procedure. In the example, three line segments are identified as not having complete topology, and are set aside in column M for further investigation.

Table 3–3. Sample output of the topological checking procedure using Microsoft Exc

A	B	C	D	E	F	G	…	
1	3	84488171	1	1	FALSE			
7	1	84488175	1	1	FALSE			
8	7	84488178	1	#N/A	TRUE	The following segment ID is not topologically consistent:		84
11	9	84488181	#N/A	1	TRUE	The following segment ID is not topologically consistent:		84
3	11	84488182	1	1	FALSE			
12	14	84488184	#N/A	#N/A	TRUE	The following segment ID is not topologically consistent:		84

To complete a topological check of TIGER file information, complete the steps outlined below using TransCAD 2.1 and Microsoft Excel 4.0.

1) After assigning a control section number and type to each of the segments along a route (as specified later in this chapter), select all of those segments using *Select* from the menu board.

2) In the data editor window:

   - toggle on the node labels for the line layer,
   - activate the *only the ID* field, and
   - write the file to a worksheet file using the *print* command.

5) Import this file into Microsoft Excel 4.0. It is best to import the first data point into cell *A1*.

6) Type in the command for row 1 in cells D through G and cell M just as in Table 3–2.

7) Copy each column down to the correct number of records imported from TransCAD.

8) Print Columns G and M for a complete listing of the segments that may not be topologically correct.

9) Using the listing of the segments, go back into TransCAD and query the database to locate the segments and examine the reason for their lack of topological consistency.

The dynamic segmentation process

Label each primary road segment with a control section number and a designation of link type. As outlined on page 24, five new fields were added in TransCAD to the existing TIGER line database to aid dynamic segmentation. For the highways to be dynamically segmented, it is necessary at this stage to label every segment of the highway with attributes for two of the five fields: *Control Section* and *Type*. An automated process was developed to facilitate labeling; this process worked for most of the primary road segments within a given county. It consists of the following steps.

1) Choose *links* as the *Active Layer* from the *TransCAD* menu.
2) Highlight all the line segments comprising each highway within the county's TIGER file on the *Map Display* screen (based on the conditions set up previously) by selecting (from the *Select* menu) *Several*, *On Condition*, and then *One condition* from the pop-up menu. Select the primary road segments by toggling their names one at a time.
3) After having selected a primary road, switch to the *Data Editor* screen. From the data fields displayed, two fields (*Control Section* and *Type)* are chosen for editing. Move the cursor to the *control section* field and choose the *Edit* option from the menu board, followed by *column formula* from the edit menu. The appropriate control section number corresponding to the selected highway/line segments (e.g., 10380 for I–380) is typed in the space where the user is asked is to *enter formula*. The same process is repeated for another data field, *Type*, where the character *T* is entered as the value for all line segments in the network as it refers to truncated line. No editing changes are made to the other fields.

The process is repeated for all the primary road segments in the county, by selecting each road as *One condition* under the *Select* menu.

Through this process, all line segments within the county with the attribute *name* matching exactly with a primary road name were labeled (e.g., segments named *US Hwy 6* and *I–80*). However, several other line segments are part of the same primary road, but are named differently in the TIGER database. This is a common occurrence when highways pass through urban areas. For example, US Hwy 6 is named Riverside Drive in downtown Iowa City. In such a situation, the user must resort to manual editing using on-screen editing functions. This process consists of the following steps.

1) Choose *links* as the *Active Layer.* To highlight all the primary roads within the county in the *Map Display* screen, select them *On Condition* (using *Union of Conditions*) on the basis of highway names.
2) Look for and zoom into areas where there are discontinuities in the highlighted roads (i.e., where highway segments have not been highlighted).
3) Once such a segment is found, select the option *Query* from the menu board, followed by *Edit* from the Query menu. After choosing *Edit*, an arrow icon is displayed.
4) Use the arrow icon to click on the line segment of interest.
5) From the data fields displayed in a form view of the entity, choose two fields (*Control Section* and *Type*) for editing. Type the appropriate control section number corresponding to the selected line segmented in the space

allocated for that field (i.e., in the example of I–380 in Linn County, control section 10380 would be entered). Enter the character *T* as the value of *Type* for all line segments in the network (this refers to a truncated line). No editing changes are made to the other fields, nor is it necessary to change the *name* of the segment because the *Control Section* is its unique identifier.

6. To leave form view, click on *Quit*.

Repeat this process for all primary road segments whose names in the county's TIGER file do not match their original highway names. If, however, the user comes across a series of such connected highway segments (which are not highlighted when selected *On Condition* of highway names), selecting all of them *by Shape* will save considerable time and effort. Choosing *By Shape* enables one to select all entities contained within or intersecting an arbitrary polygon that is drawn on the screen (in this case around the connected highway segments mentioned above). To accomplish this, the user completes the following steps.

1) With *links* as the *Active Layer*, choose *Select* from the menu board, *More…* from the *Select* menu, and then *by Shape*.
2) To draw the "arbitrary" polygon on the screen, press the <left> arrow button to start the polygon.
3) Press the <right> arrow button to locate any number of intermediate points on the shape.
4) Press the <left> arrow button to end the shape.

All entities in the *links* layer that are contained in or intersect the specified shape are selected and highlighted. These are the segments where control section numbers and line types should be added. Switch to the *Data Editor* window and use *column formula* to label the two fields with appropriate attributes, as discussed previously.

After control section numbers and line types have been assigned to all highway segments within a county, set up a new set of "conditions" using the control section numbers, in addition to the existing conditions comprising highway names (see Section D). Once in the Conditions window, the user selects *Control Section* from the *Data Field* column, and *Equal to* from the *Operator* column. The control section number corresponding to the highway is entered in the column *Value*. In subsequent stages of the project, the primary roads within the county will be highlighted in the *Map Display* window by selecting (*On Condition*) the control section numbers, not the highway names.

Prepare input milepost data and add values to the Atlas database using procedure MPTAG (input file). The MPTAG procedure automatically assigns *start Milepost* and *end Milepost* values and *CS/Sequence* numbers to segments in the Atlas database based on data from the Paradox comma-delimited text file. An input file must be prepared using a word processor or text editor. The input file must contain one row of data for each continuous series of segments in the line database that have the same *Control Section (CS)* number and *Type* (i.e., *T* for truncated line). Each row of data indicates known milepost locations along the control section.

Rows in the input file have the following format:

$$ID, Type, N, Node_0, MP_0, \ldots Node_n, MP_n$$

where:

ID	is the control section identification number,
Type	is a type identifier (*T* representing truncated line),
N	is the number of known milepost locations and values,
$Node_n$	is the node ID for the nth node, and
MP_n	is the milepost value for the nth node.

Milepost values must be listed in increasing order within each row of the input file.

The $Node_n$ and MP_n specifications are not necessary unless the user wishes to examine only a small section of a street rather than the whole segment.

An example of this input file specific to the project is developed for the Linn county I–380 study. This example text file corresponds to the format above with data in the following format:

10380,T,1,10367,0

This line in the input file indicates that control section *10380* starts at node number *10367*, where the milepost value is *0*. By default the MPTAG will determine all subsequent milepost values based upon the actual length of each segment in the line database.

The data for the input file are determined in the following way:

1) *ID* is the control section number of the primary road segment to be mileposted. This number can be retrieved from the Paradox file or from the data editor in TransCAD.
2) *T* is the TIGER file specification for a truncated line for linearly segmented objects.
3) *N* simply specifies the number of known milepost numbers along a particular primary road segment. It is easiest to specify one known node because TransCAD will assign the rest in the MPTAG module.
4) To find the *Node* number, activate the node layer of the TIGER file database and click on the first node along a primary road segment. Because data for road segments in the Iowa DOT base records run east/west and north/south, the first node number specified should be the node number that begins at the west or south end of a primary road segment.
5) *MP* is the milepost value at the beginning node of the road segment and can be determined by viewing the paradox file to see the first milepost number for each primary road segment.

The input file contains one row of data (as above) for each primary road segment. Each input file can contain as many rows of data as desired. In other words, more than one primary road segment can be atlas-mileposted at a time. Also, it is important to note that it may be necessary to divide a road segment up due to breaks in the road or overlaps with other primary road segments. In these cases, a primary road segment may have two or more rows of data dedicated to it in the input file. This type of occurrence is described in the following example.

In Iowa City, Iowa, two primary road segments overlap (i.e., they are the same road) in the downtown area. In such a case, there are two rules that must be followed in assigning *control section* or *ID* numbers:

1) the road segment with the higher functional classification takes precedence and
2) if the two primary road segments are of the same functional classification the one with the lowest number identifier takes precedence.

So in the aforementioned example, U.S. Highway 6 takes precedence over State Highway 1 because it has a higher functional classification. The *control section* and *ID* value assigned to the road segments where the two overlap would correspond to that of U.S. Highway 6 (if both roads were state highways, the one with the lowest numerical identifier would have its *control section/ID* assigned in the overlap area).

In cases such as this, gaps in the *control section* number will occur for the road segment with a lower functional classification (State Highway 1, in our example). To milepost that road segment correctly, then, it is necessary to have one input line for each continuous portion of the road segment.

For the *first input line* of that road segment, *from* and *to* nodes must be specified because the atlas mileposting will not run continuously from the first node all the way to the last node of the road segment. The *End MP* of the last segment of the highway (where it stops and the overlap starts) is noted (with the help of a *Query*) and is specified in the first input line. This is the milepost value for the ending node of this half of the road segment.

For the *second input line* of the same road segment, the node number represents the node where the overlap ends and the road continues. To compute the milepost value, measure the distance from this node to the one where the overlap begins (i.e., the total length of the overlap):

1) Select *Query* from the menu board, followed by *Distance*, to bring up an arrow icon on the map.
2) Using the mouse, point at one end, and stretch the resulting line along the overlapping segments.
3) At the other end, press <Enter> to view the distance between the starting and ending point. The milepost value for this half of the highway segment is the sum of this distance and the ending milepost value from the previous input line.

For example:

10380,T,2,13067,0,13081,13.31

10380,T,1,14000,18.5

In this case, the first portion of *control section* 10380 will be mileposted between *nodes* 13067 and 13081 with values of 0.0 to 13.31; the second portion of *control section* 10380 (starting at node 14000) will start at milepost value 18.5 (the measured distance or length of overlap being 5.19 miles, the difference between 18.5 and 13.31), with subsequent milepost values determined by the length of the segments in the Atlas database.

Once the input file format is completed, the following procedure is implemented to atlas-milepost the road segments.

1) Return to TransCAD.
2) Activate the line layer of the database by choosing *Active Layers* from the *TransCAD* menu.
3) Click on the *Procedure* menu and select *Choose*.
4) Select *Dynamic Segmentation* from the *Procedure Classes* menu.
5) Select *Atlas Mileposting* from the *Dynamic Segmentation* menu.
6) Choose the input file that contains the milepost values.

Once this process is complete, add the *Start MP*, *End MP*, and *CS/Sequence* fields to as many records as are specified in the Atlas database. To ensure all road segments have been properly tagged, the data editor can be used to review the updated *Start MP* and *End MP* data. The *CS/Sequence* field can be used to display all segments in a control section in order. Also, the user can simply query along segments of the primary road in the *Map Display* windows to determine whether the *Start MP* and *End MP* have been properly added. Further, *MPTAG* produces a text file that lists any errors detected during the procedure.

Create Milepost Attribute Tables (MATs) with data describing the network characteristics at specific milepost locations. MATs are data tables containing milepost-referenced data on one or more attributes of the data. The MATs contain data that describe the characteristics of the network at specific milepost locations.

1) TransCAD stores MATs in a proprietary format (not a TransCAD database) to speed up the dynamic segmentation process.
2) Each MAT describes characteristics of facilities at different locations.
3) Each MAT is created from an input data file containing one line that describes attribute values at the beginning of one route, and another line for each location along each route where the attribute values change. A variety of MATs are created for a route, each containing a different set of attributes.

Procedure MATBUILD. MATBUILD reads route/milepost-based data from a fixed format or comma-delimited input file and creates a milepost attribute table for use in the dynamic segmentation process. Two text files are used as input to MATBUILD. The first file contains attribute data for one or more attributes at specific milepoint locations. The second input file contains information used by MATBUILD to figure out where the required input data can be found in the first input data file.

Comma-delimited input files were used to build MATs from the Iowa DOT base records attribute data. The first file is the data file that contains Route and Milepost data for control section 10380. This file is obtained from Paradox and contains all the appropriate attribute fields for a given county in a comma-delimited format. Because the example deals with I–380, retrieve the Linn County Paradox database. As mentioned in an earlier section, this file was written to the *c:\gist_txt* subdirectory. To create the MATs, the file is combined with the second text file describing the format of the file. The following is an example of the structure of the data file:

10380,0.0,918,17737

10380,1400,930,17504

10380,1800,945,17600

The first column in the data file contains the control section number under study. The second column is comprised of the starting, ending and intermediate milepost numbers along the control section where attribute information changes. The first row, column two for each control section has a milepost value of zero, or the keyword START. The last row, column two for each control section has the highest milepost value on that route, or the keyword END. All routes and control sections must have at least two input lines. The third and following columns contain attributes for the particular segment. In this case, column three represents *Number of Motorcycles* and column four represents *Number of Automobiles*. Interpreting the above file shows that control section 10380 has 918 motorcycles and 17,737 automobiles on average traveling between mileposts numbered 0.0 and 1.4 (the rest of the file can be interpreted similarly). It is important to note that the two columns of attribute data are used only as an example and that the actual Linn County Paradox file contains 178 attribute fields after the milepost column.

The second file is a format file which ensures that TransCAD will search the correct paths for attribute information. The following is an example:

c:\gist_txt\linn.txt

1

2

"Motorcycles",3

"Automobiles",4

etc.

The first line describes the path where the data file can be found, the second indicates which column the control section is in, and the third indicates the column in which the milepost information can be found. The fourth and subsequent lines describe the attribute fields to be created and indicate in which column of the data file information regarding the attributes can be found. This file usually contains the file extension *.in* and should be named and placed in a meaningful subdirectory (e.g., *c:\gist_txt\in_file\linn.in*).

Once these files have been specified, the MATBUILD procedure in TransCAD can be implemented to create the milepost attribute tables. The following instructions describe how to implement MATBUILD in TransCAD.

1) Click on the *Procedure* menu and select *Choose*.
2) Choose *Dynamic Segmentation* from the *Procedure Class* menu.
3) Select *Build Attribute Table* from the *Dynamic Segmentation* menu.
4) Specify *comma-delimited* file at the *Input File Type* menu.
5) Enter a name for the MAT to be created and write it to a meaningful subdirectory (e.g., *c:\gist_txt\att_tbl\linnmat.txt*).

6) Select *Direction-forward* (because the data file starts with the first milepost).

7) Enter the name of the format file that describes the input attribute data (e.g., *c:\gist_txt\in_file\linn.in*).

At this point, TransCAD will begin building the milepost attribute tables. When the process is complete, TransCAD will indicate the number of records and corresponding fields that have been processed. The user can check the MATBUILD process by seeing whether the number of records and fields processed match exactly the number of records and fields in the Paradox base file.

Group data fields associated with Atlas by creating text files that list field names and associated MAT file names. To implement the dynamic segmentation procedure, there must be a text file with the extension *.dsg* that lists the field names and file names of all MAT fields. There can be many such files, each containing lists of related data fields. This file tells the program where to look for the information needed to complete the dynamic segmentation routine. The following is an example of such a file:

> *"Dynamic Segmentation Data Fields"*
>
> *Motorcycles,c:\gist_txt\att_tbl\linnmat*
>
> *Automobiles,c:\gist_txt\att_tbl\linnmat*
>
> etc.

This file should be kept in a meaningful subdirectory (e.g., *c:\gist_txt\dsg\linn.dsg*). At this point, the user is ready to return to TransCAD and invoke the dynamic segmentation procedure.

TransCAD's dynamic segmentation procedure. This procedure creates a line database that is dynamically segmented based upon a set of attributes chosen from MATs. This work database has the minimum necessary segmentation required to identify all homogeneous line segments. The following instructions explain the procedure for implementing dynamic segmentation in TransCAD:

1) Activate the line layer in the Atlas database by selecting *Active Layers* from the *TransCAD* menu.

2) Select the Atlas segments to be dynamically segmented using commands on the *Select* menu.

3) Choose *Several* from the *Select* menu, then choose *On Condition*, and specify *Union of Conditions* from the pop-up menu.

4) Activate the control section numbers for all primary road segments by toggling the boxes.

5) Next, select *Choose* from the *Procedure* menu.

6) Select *Dynamic Segmentation* from the *Procedure Class* menu.

7) Select *Dynamic Segmentation* from the *Dynamic Segmentation* menu.

8) Enter a path and file name for the database to be created (e.g., *c:\database\linn_dsg*).

9) Choose the *.dsg* file that contains the list of MAT data fields and the names of the MATs in which the fields are found (e.g., *c:\gist_txt\dsg\linn.dsg*).

10) From the pop-up menu, choose the attribute fields on which segmentation should be based. This can be done at the user's discretion; all or a subset of the fields may be chosen for dynamic segmentation.

11) Once it has completed the process, TransCAD will alert the user to any errors that occurred during dynamic segmentation.

12) The new dynamically segmented database can be added by choosing *Application—Add Database* from the *TransCAD* menu.

13) Activate the new database by choosing *Segments* from the *Active Layer* of the *TransCAD* menu.

The resulting database can be used like any other TransCAD database. Conditions can be set to perform attribute queries; themes, labels or bandwidths can be used to display attribute data; and detailed information can be extracted using the data editor and statistics windows. In conclusion, dynamic segmentation is an efficient, non-repetitive way to manage linear data within the realm of a GIS.

ALTERNATIVE SOURCES OF BACKDROPS FOR TIGER FILE UPDATES

Digital line graphs

Advantages

1) They are available from the local division of the USGS (although coverage is limited).

2) They have been used in the field of TIGER file updating, therefore contacts can be made for consulting purposes.

3) The scale of the line graphs (1:24,000) is better than some alternative sources.

Disadvantages

1) Conversion of *DLG* files into TransCAD-readable format is a complex and convoluted process that can take weeks to complete.

2) County coverage of *DLG* maps is incomplete and unavailable from the USGS.

3) Members of both the private and public sectors question the accuracy of *DLG* files. Many believe a more accurate source should be sought.

4) The *DLG* files are large and will probably be hard to deal with in TransCAD. We obtained *DLG* files for four sections of Linn County (i.e., including only partial coverage of the county). There were 44 files with an average size of two to three megabytes each.

Scanned orthophoto quads

Advantages

1) They are capable of producing aerial photography that represents ground truth more accurately than previous aerial photo techniques.
2) They can be scanned as a backdrop that is directly readable into TransCAD.
3) They are a higher level representation of road segments than any other form of cartographic backdrop.

Disadvantages

1) Full coverage of a given county is not available in this format.
2) They are divided into small sections of coverage (i.e., sections of coverage that are one mile by two miles). Scanning numerous orthophotos is therefore necessary if full coverage of a county is desired (and if the orthophotos are available).
3) Scanning can cause some distortions in the map that will be translated into TransCAD.

Iowa DOT AutoCAD DXF files

Advantages

1) Full coverage is available for the five-county study area.
2) Initial translation of these files into MapInfo format indicates that the files appear to be a very close representation of the hard copy base maps sent from the DOT.
3) *DXF* files contain 63 layers of information that may be useful in future applications of the project.

Disadvantages

1) Several problems have occurred in the translation of *DXF* files to TransCAD. From our experience, the module that performs this task appears to be underdeveloped.
2) Accuracy of the street line representations of the *DXF* file in TransCAD is worse than in TIGER files. Interestingly, however, this is not the case in MapInfo. This further leads us to believe that the TransCAD *DXF* translation module is the cause of the inaccurate representation.

Digitizing from Iowa DOT base maps

Advantages

1) Digitizing is a quick and easy way to fix areas in the TIGER files where representation of the road segment is flawed. The Iowa DOT base maps would be used as the digitizing source of accuracy.
2) This method may be more efficient than translation processes, which are often very complicated and inefficient.

3) Iowa DOT maps are readily available for the digitizing process.

4) Digitizing capability is built into TransCAD; therefore it will be possible to redigitize the TIGER files directly.

Disadvantages

1) Digitizing directly from the base maps assumes that they are an error-free source.

2) Human flaw can result in errors in the digitization process and in the resultant TIGER file.

Digitized orthophoto quads

Advantages

1) The digitizing of some orthophotos has been taking place locally (Johnson County Auditor) and the resulting files are available.

2) They are highly accurate representations of road segments (accurate within ten to 15 feet).

Disadvantages

1) The digitizing of the photos is a time-consuming process and therefore full coverage is not available. It will be two to three years before all of Johnson County is represented in digitized orthophoto format.

2) They are being digitized into AutoCAD DXF file format. As mentioned earlier, this is problematic because the TransCAD *DXF* translation appears to be flawed.

CHAPTER 4

TOWARD A CONCEPTUAL FRAMEWORK FOR THE VISUALIZATION OF NETWORK INFORMATION

A network is an important, though often implicit, data model that guides our general conceptions of geographic space. Notions about communication, interaction, and flow are as basic to geography as those of space and place, and serve as important foundation elements for the fundamental themes of geography (Joint Committee and Geographic Education 1984). Networks serve to facilitate, inhibit and spatially organize movement, and thus are critical components for understanding many geographic phenomena, be they human or physical. Knowledge concerning the movement of people, goods and ideas through transportation networks is a basic component in a large class of optimization and location-allocation models, in most social communication and epidemiological diffusion models, and in many economic models and models of regional settlement structure.

Despite their general importance, there are few well established cartographic conventions that concern general network representation. Dent (1996), for example, provides a detailed discussion of flow maps and dedicates a few additional pages to examples of line symbols. Network representations, however, are not discussed explicitly. Furthermore, he recommends only five readings on flow maps, in comparison to an average of 29.47 additional readings recommended in other chapters. His fourth edition (1996) devotes slightly less space to the same topics. Robinson and colleagues (Robinson and Sale 1969; Robinson et al. 1995) provide a similar, brief treatment of this subject. Nor is the topic discussed in any depth in current automated cartography texts (cf. Clarke 1995; Cromley 1992). Bertin's (1973, 1981) treatment of networks (réseaux) is highly abstract and, although cited, is rarely used as a source of production guidelines. The need for computer-supported network visualization and network-based modeling is increasing, driven by the development of GIS-based analytical techniques and intelligent transportation systems (ITS). The absence of an explicitly articulated framework for network representation is therefore problematic.

In this chapter we begin to articulate a basic conceptual framework that can distinguish between different types of network and interaction maps and will serve as a guide in choosing appropriate network visualization techniques. The framework encompasses symbolization, map types, and base-map transformations appropriate in different problem contexts. Our definition of networks and network types is generous and inclusive, embracing everything from social networks, to highways, to networks of radio beacons. The principles that guide the framework focus primarily upon the purpose of the representation and its role in communication rather than the spatial properties of the network in isolation or its internal digital representation. First, we present some examples of the special difficulties encountered when networks must be rendered cartographically. We then outline the fundamental dimensions of the framework.

NETWORK MAP TYPES

To illustrate the shortcomings of current network representation strategies commonly adopted by GIS software and other automated visualization packages, we have chosen three examples of map types involving important cartographic operations that should be included in the general repertoire of network visualization techniques. These operations are not easily performed using existing GIS-based visualization tools. Nevertheless, they (and undoubtedly others) illustrate alternative base-map transformations that are often of interest when designing network maps. The following map types are modifications of the base map.

- **Flow maps.** The base map is modified based on attribute values of the network. In many cases, the line width is also modified based upon the same attribute values.
- **Connection maps.** The purpose of this map is to guide general navigation through clearly defined networks.
- **Sketch maps.** The base map is modified based on the salience of different portions of the region or network. Frequently these maps are used to guide navigation to specific destinations.

Flow maps

Flow maps are probably the most extensively developed type of network map found in cartographic texts. A stylized map, as seen in Figure 4–1, can easily be produced using existing GIS software. It has a conventional planimetric base-map projection and the amount of flow is indicated by line width (color and line type are the other easily available options for nominal and ordinal variables; e.g., Morrison 1971). Because of the spatial distribution of flow intensity, however, there is a considerable amount of overlap among the line symbols in congested areas. Consequently, this map is of little use if one is interested in the amount of traffic moving between two locations in such congested areas, perhaps with the goal of mitigating a specific problem, rather than gaining a broad, general, or regional-scale impression of traffic density. The map shown as Figure 4–2 attempts to address this problem by radically transforming the base map on which the flows are plotted, allowing each line sufficient space to be clearly visible. The problem can sometimes be solved by changing either the range of line widths or changing the geodetic projection of the base map, though not in all cases is it solvable (e.g., Tobler 1987, p. 159; Monmonier 1993, p. 179). We refer to this process as an attribute-driven base-map transformation because it is the value of the attributes linked to network elements that drives the transformation of space. This type of transformation is a network equivalent of a polygon-based cartogram base-map transformation (Kadmon 1982).

Although automated flow mapping is a recognized area of research, progress has been punctuated by long periods of inactivity. Tobler (1987) wrote a computer program to automate migration mapping and discusses important design issues including the choice of line types and symbolization. Kern and Rushton (1969) and Wittick (1976) are other examples of early computer-based mapping efforts. More recently, Asproth et al. (1994) discuss stream (flow) mapping in a GIS context. Although not directly dealing with the construction of flow maps, Clark (1977), who developed an algorithm to build time-distance transformations of transportation networks, illustrates many of the technical difficulties of such a process. The bulk of research in 2- and 3-D flow

visualization, however, concerns fluid flow and is being conducted outside of cartography and GIS, in such major laboratories as the following:

- MIT's Department of Aeronautics and Astronautics (http://raphael.mit.edu/haimes.html),
- Deutche Forschungsanstalt für Luft- und Raumfahrt (http://www.ts.go.dlr.de/sm-sm_info/STinfo/STgroup.html),
- Lawrence Livermore National Laboratory (http://www.llnl.gov/graphics/sciviz.html), and
- NASA's Langley Research Center (http://www.icase.edu/docs/hilites/index.cs.viz.html).

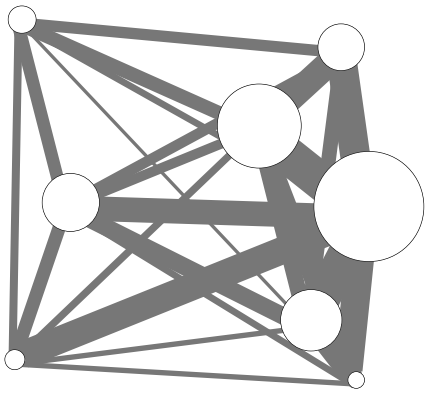

Figure 4–1. Stylized flow map

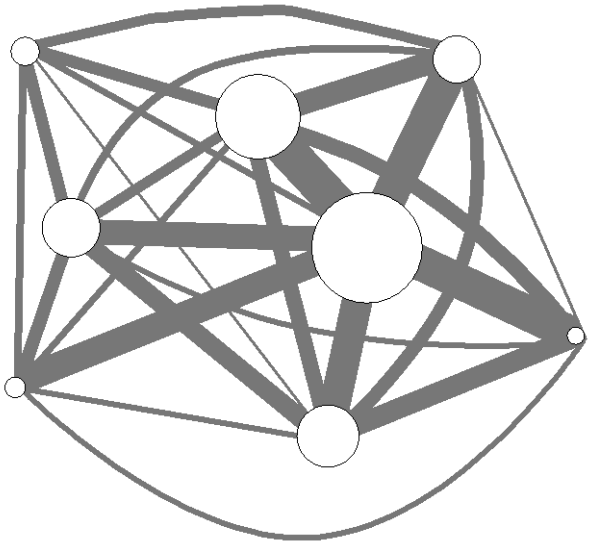

Figure 4–2. Attribute-driven base-map transformation to alleviate overlap of line symbols in congested areas

Connection maps

Until the 1930s, the London Underground map attempted to locate stations and routes in their correct geographic location (Vujakovic 1990, p. 45). This type of network representation strategy is directly supported by most GIS software. It was only after considerable resistance that the London Passenger Transport Board introduced a new and transformed network map that emphasized the *connections* between routes. This map has since attained an iconic status in its own right. This representation strategy was successful and has consequently spawned a host of imitators (see examples in Vujakovic 1990; Kadmon 1982), many of which are being made available digitally in a way-finding context (http://metro.jussieu.fr:10001). These types of base-map transformations are different from the first example in that legibility and other cartographic and visual considerations drive the base-map transformation instead of network attribute values (Kadmon 1982, pp. 6–9; Monmonier 1991, pp. 34–35).

Network sketch maps

The map shown in Figure 4–3 is a conventionally projected network map that shows the area around Gig Harbor, Washington. Figure 4–4 is an advertising map that illustrates the path a traveler is advised to take to reach a specific restaurant. Approximately the same geographic area is shown in both maps, extending north from the Interstate 5/Route 16 interchange to the top of the harbor. The main difference is that in the map on the right, more detail (and a larger area) is devoted to those parts of the map that are complex (involve more driving decisions) and are closer to the destination. This structure is common to both cognitive (or mental) maps and hand-drawn sketch maps (Lynch 1960; Gould and White 1974, p. 139). The baseline transformation involved in constructing this map is different from that of the connection map (e.g., the London Underground map) in that it is a rubber-sheet transformation of the area and not a simple graph-theoretic transformation of the network; important geographic landmarks such as the harbor have been transformed along with the road network. Monmonier (1991, pp. 59–66) discusses some important characteristics (both persuasive and perverse) of this map type.

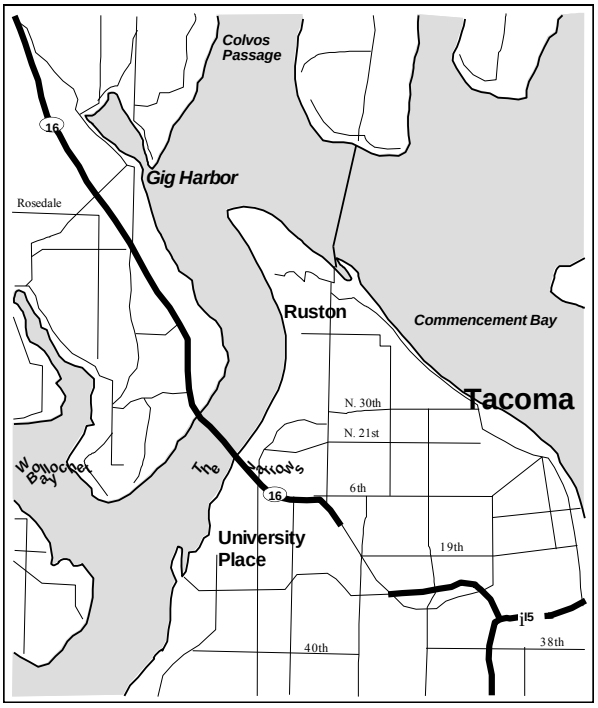

Figure 4–3. Conventionally projected network map

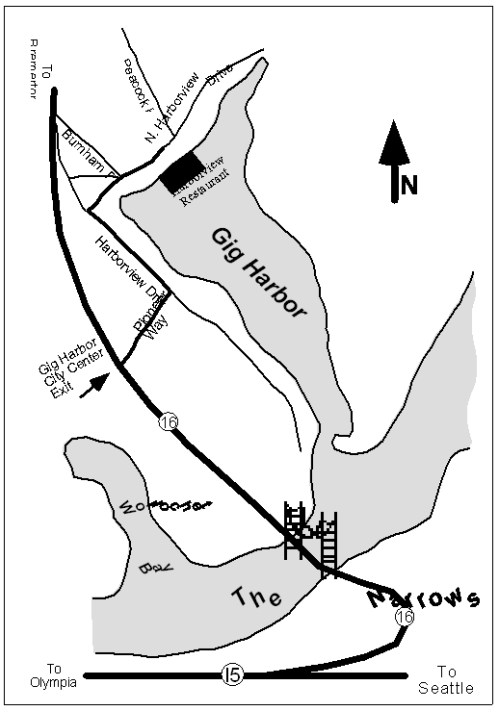

Figure 4–4. Advertising map providing directions to a specific location

The difference between connection maps and network sketch maps can be better understood if we consider the network structures involved and navigation tasks supported. The obvious difference, that one is designed to support general navigation (many origins, many destinations) and the other usually to support guided navigation (many origins, single destination) does not completely explain the choice of a topological or rubber-sheet transformation.

In the connection map example, the network has a relatively “inflexible” structure, with a limited number of links connecting a given set of points of entry, transfer, and exit. The navigation task is similarly simplified, consisting mainly of choosing the necessary links between an origin and a destination. In these systems, once a network link is entered, if a mistake is made, there are no opportunities for attempting a course correction until the next “station” is reached, either because of the physical structure of the network (e.g., subways, railroads, and interstate highway systems) or because the navigator is not in control of the vehicle (e.g., bus systems). Under these circumstances, the value of off-route objects or landmarks as navigational cues is minimal. In subway systems, there are rarely any visual cues to indicate when one has chosen an inappropriate link (beyond the stations themselves), so it is not entirely unexpected that subway maps are characterized by the most extreme topological base-map transformations.

Network sketch maps, on the other hand, usually show flexible networks, with many points of entry and exit, and many possible alternative routes between any given origin and destination. The general navigation task is correspondingly more complex. We have already mentioned that network sketch maps are designed to provide directions for navigators from an unknown number of possible origins to a single, specific destination that is usually new to the traveler. This explains the greater amount of space dedicated to the area surrounding the destination, but again, it does not explain the choice of a rubber-sheet over a topological base-map transformation. What does explain this choice is the importance of off-route cues in navigation. Off-route cues confirm that one is on the correct route, and prepare one for upcoming navigational choices (especially in complex or unknown environments). These characteristics are of greater importance in flexible networks. Off-route cues are better preserved using a rubber-sheet base-map transformation.

The relative importance of off-route cues in navigation is not confined to the distinction between flexible and inflexible networks. Personal navigation styles and the familiarity of the route traveled can also play a role. Understanding these issues can possibly explain the contradictory results concerning preferred road map type reported by several studies (Sheppard and Adams 1971; Astley 1969).

ELEMENTS OF THE FRAMEWORK

Network characteristics

The type of network to be represented plays an important role in determining an appropriate representation strategy. It should certainly not be assumed that a network is a simple combination of lines and points that can be satisfactorily represented using only a knowledge of line and point symbolization heuristics. A network is more than a collection of lines and points. At the very least, a network is an *organized* collection of these cartographic objects, and is often partially defined by co-bounding polygons as well (for example, see Cromley 1992, p. 84). This is not to imply that representing a network is a simple matter of correctly depicting topology. Other important characteristics of networks that influence their representation are:

- the degree and kind of organization present in the network,
- the nature of flow and interaction supported by the network,

- dynamism,
- the nature of fixed infrastructure that exists independently of actual flow, and
- the network's relationship to its environment and context.

When networks are hierarchically organized, the well-defined importance of individual network elements can be used to guide cartographic choices such as symbolization. In other cases, networks are heterarchical, a condition in which the relative importance of network elements is unclear, and may depend on the exact state of the network as well as the context and purpose of the representation. In such cases the relative importance of network elements may still be used to guide the network representation, but choices are sometimes ambiguous and may require careful thought and some preprocessing. Also, the complexity of the hierarchy or heterarchy and the number of levels it possesses may further direct a particular cartographic representation. For example, certain hierarchies may be nominal or ordinal in nature, while others may have interval, or even ratio relationships between levels. The number of levels the hierarchy possesses and their relative importance will also influence generalization and simplification processes that can be applied to it.

The elements that flow or travel through a network may be discrete (e.g., automobiles) or continuous (e.g., water or traffic). Representing the distinction is sometimes desirable and thus can influence symbolization (such as scale, generalization and choice of base map). For example, if the movement of individual vehicles through the road network needs to be mapped (to indicate the location of emergency vehicles or individual buses in a system, for example), the scale, base map, and line width of the roads must be designed to accommodate the symbols for individual units both in size and quantity. These design criteria will impose constraints on the form of the final map that would not apply if the movement of vehicles were to be conceptualized in terms of a flow volume, density, or frequency of service. The amount of flow the network supports and the variability of that flow may also influence the design of a map, especially animated maps.

The network infrastructure itself may be somewhat dynamic. Certain network attributes along links are open to change (e.g., detours, periodic one-way streets, or speed limits) and allowances should be made if the desired base-map transformation relies heavily upon network attributes. If the changes are predictable (such as rush hour traffic) it may be best to design the base-map transformation based upon maximum attribute value. If the network attributes are unusually dynamic and unpredictable, the increased legibility of the transformed base map may be outweighed by the difficulty of dealing with radically different maps at different times, and a more planimetric representation strategy may be indicated. Furthermore, not all networks, not even all transportation networks, are comprised of a relatively fixed set of invariant linear links such as roads. Nor are all networks under the control of a single planning body that determines the further development of the infrastructure. In some networks it is the very interaction, or demand for interaction, between places that creates a connection; should the pattern of interaction change, the network will adjust accordingly (e.g., the Internet or airline networks). In other networks, more often communication networks, the "links" over which interaction occurs are more properly described as polygons than lines (e.g., cellular phone networks) or some combination of linear and areal links (e.g., phone service with both wire and satellite connections). In all of these cases, the network links may

experience periods of dormancy when interaction is not actually occurring, and periods when, while there is no physical spatial expression of a link, it still exists potentially or legally. The spatial expression of links may also vary over time (e.g., riverbeds and satellite-based systems). Given different problem contexts, some or all of these dynamic states may need cartographic representation, a task that becomes more critical when creating digital base maps designed to support automated, real-time, or interactive displays.

Another important characteristic of the network is how it articulates with, or fits into, with the surrounding environment. A network with a limited number of fixed entrance points (e.g., a subway or interstate) will present different constraints and opportunities than one with a greater number of fixed entrance points (e.g., a bus system or county highways) or than a multitude of very densely distributed points of entry (e.g., a city street network). Similarly, a network that is isolated from the surrounding environment by a limited set of navigational choices (e.g., the subway system) will admit different representational strategies than one where there are many navigational decisions or where off-network cues play a role in navigation (e.g., the city street network). In the case of the isolated network, a connection map strategy could be adopted, while in the other cases a network sketch map approach is more appropriate.

Cartographic elements

The types of fundamental cartographic elements that require symbolization is another important element in network visualization. These cartographic elements go beyond the simple and commonly used “points, lines, and polygons” mnemonic in that different types of points, lines, and polygons involve different treatment. For example, the widely promulgated digital cartographic data standard (see *The American Cartographer* 1988 for the proposed standards) recognizes several types of points associated with networks (see Table 4–1 for an abbreviated list), each of which may require special treatment and symbolization. The points associated with “events” in dynamic segmentation (see Dueker and Vrana 1992) pose special difficulties because they capture the most dynamic and variable characteristics of the network. The lines and polygons associated with a network, like points, also represent different functional elements. Polygons, for example, may represent a feature that determines the form of the network (e.g., a watershed), a feature whose form is modified by the network (e.g., a sales area), or may in fact be the link of communication or movement over which the network operates (e.g., satellite transmissions or radio transmissions). These different roles should be clearly distinguished in any network representation. Using polygon fill, for example, to symbolize the area of radio beacon coverage is only one potential strategy, and will not always be the most effective option. Furthermore, some combinations of cartographic elements will influence other visualization decisions. Finding ways to automatically recognize the number and nature of such interfering goals and symbolization is an important technical issue that will require further research, especially as real-time automated systems are constructed. Understanding the different kinds and roles of network elements possible and clearly defining the purposes of any network representation may aid in this task.

Table 4–1. Classes and sub-types of network elements

Points	Lines	Polygons

/destination ction/ hipment es/shape	• May or may not be the medium of movement (e.g., may represent a barrier to movement) • May be either directed or undirected links	• May be a factor influencing or creating the network (e.g., a watershed) • May be influenced by or the result of the network (e.g., a retail sales area) • May be the medium of movement (e.g., the area of radio beacon coverage)

Establishing conceptual links between typologies of the phenomena, the relevant data models, and their graphic representation (proposed by Jenks 1963, 1967) is a profitable approach for providing guidelines in design strategy. The best known example of this is for spatially extensive (continuous) variables and is from MacEachren (1992, Figures 8 and 9, p. 16), although Bertin (1973, 1981) makes use of generally similar concepts. MacEachren's typology emphasizes two important axes of variation in spatially extensive variables: whether they are continuous or discrete in nature, and whether their variation is abrupt or smooth (see also MacEachren and DiBiase 1991; MacEachren 1995 pp. 302–304). A similar typological structure will need to be developed for important characteristics of network elements. Potential axes of variation (many of which we have already alluded to) include whether the network and its elements are:

- diffuse or focused (an attribute similar to fuzziness and the abrupt-smooth axis of variation in the spatially extensive variable typology),
- continuous or intermittent in time,
- fixed or indeterminate (is their existence independent of or contingent upon interaction),
- discrete or continuous in space (essentially similar to the discussion of "flexible" and "inflexible" network structures), and
- spatially persistent or spatially dynamic.

Many of these axes are not independent, and further research is needed to identify any simpler, critical characteristics implicit within this list.

Cartographic resources

The fundamental resources available for network visualization are basically those available to traditional cartography. Bertin (1973) was the first to formally propose what have come to be known as the visual variables, variables that have been systematically investigated and expanded since then (MacEachren 1995, pp. 269–290). Animation and hyperlinks are significant additions to the repertoire and are popular areas of visualization research (MacEachren and DiBiase 1991). The processes of line generalization (e.g., simplification or displacement) are not yet satisfactorily automated, but are also active areas of research (Buttenfield and McMaster 1991; McMaster and Shea 1992). Most research into generalization situates the problem in the context of scale changes, but the techniques could easily be used or modified to support generalization performed because of map purpose. This research could also prove of value when attempting to automate topological base-map transformations. While changes in scale and orientation are similarly made easier in the digital environment, they are not in themselves unique to it and

should not be naively accepted as benefits bestowed by the digital environment. In theory, base-map transformations are easy to perform in the digital environment but, apart from the choice of projection, this ability is rarely used for purely cartographic purposes and it is not well supported by most GIS software.

Base-map transformations

Base-map transformations that are typically applied to network maps can be classified along two axes of variation. One axis, which we refer to as geographic, includes the continuum of transformations of space as modified through standard geodetic and rubber-sheet projections. The second dimension, in contrast, considers the network's topological structure and how it may be altered (primarily through breaking and the omission of elements). Monmonier (1991, p. 35) refers to such operations as geometric generalization. Geographic transformations are indicated along the y-axis in Figure 4–5, while network topological transformations are placed along the x-axis.

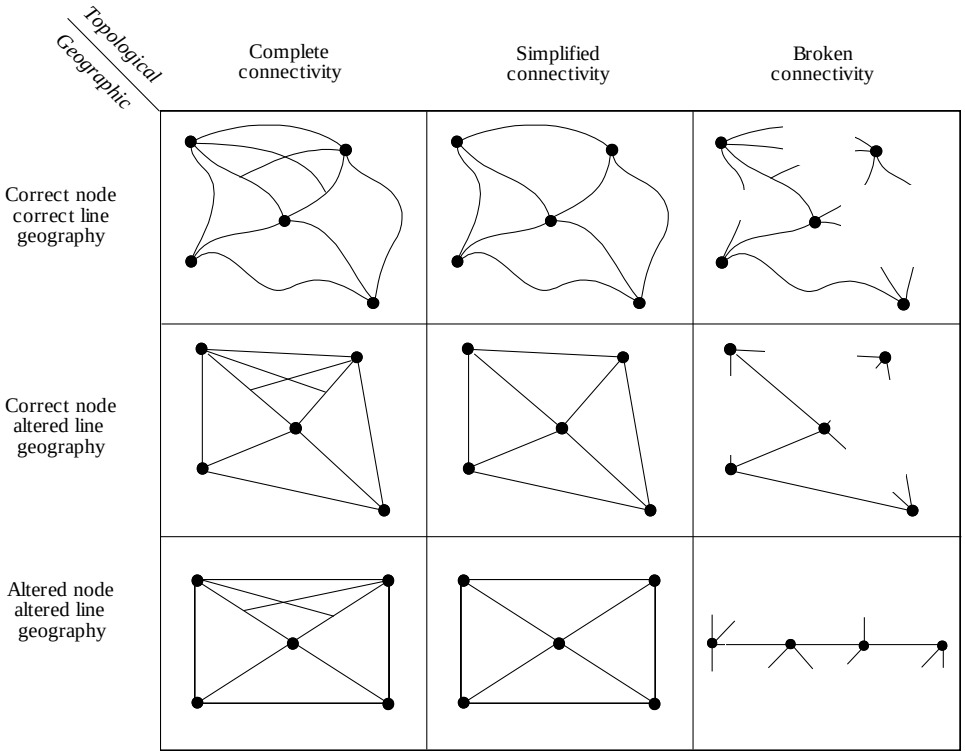

Figure 4–5. Geographic and topological base-map transformations

The map types drawn in this figure represent discrete archetypes that are derived from the associated combination of geographic and network transformations. In practice, the transitions along each axis are more continuous than indicated. For example, a map with true "correct point and line geography" as well as "complete connectivity" would possess "perfect" and "complete" detail as well as "perfect" georeferencing and is only theoretically possible. All maps and models involve some degree of generalization and abstraction. For example, most general highway maps attempt to locate nodes correctly, have at least a slightly simplified connectivity (some roads are not shown) and have

mildly altered line geography (due to generalization). In comparison, a map of the interstate system would have a more simplified connectivity in that many minor roads might be omitted. Finally, the London Underground map with its very simplified connectivity (generalized routes, rather than the exact geometry of the tunnels taken) and altered node and line geography corresponds to the archetype in the middle of the bottom row. The Gig Harbor sketch map with its broken connectivity (only the route of interest is indicated) and its altered node and line geography corresponds to the archetype in the lower right. A more extreme form of this archetype of interest to transportation planners is the linear strip map (MacEachren 1986).

The “floating” maps operationalized in many ITS systems, maps that show the immediate region surrounding the vehicle and the route to be followed conceptually belong to the column indicating broken connectivity in Figure 4–5 because they usually do not show the entire route at a single glance (although many can be toggled to provide such a view). Systems that maintain a constant orientation for the floating maps behave more like the archetype in the upper right corner where correct node and line geography are maintained. Systems that re-orient the floating maps so that forward motion is always at the “top” of the map are essentially automated strip maps.

Area-networks (e.g., radio beacons) are less common than linear networks, but could be organized into a similar structure, although the relative importance of the different archetypes would undoubtedly be different and the interpretation of the axes might be slightly different (shape, for example, would have to be incorporated into the structure). The equivalent structure for area networks could serve for many areal-based cartographic entities in that it would basically organize choropleth maps and different types of cartograms into a single framework. For example, discontinuous cartograms would correspond to the column with broken topological connectivity, while Dorling’s (1993) research into automated cartograms based upon circles would probably conform to the altered node, line (and shape), and simplified connectivity archetype. The areal equivalent of a strip map could very well be a geographically organized small multiple as presented in MacEachren (1995, Figure 9.2, p. 406).

Network map queries and their cartographic outcomes

The types of information a map is intended to provide (its problem context) influence its representation. This does not change when maps are queried interactively and may, in fact, become more problematic in a data-rich digital environment because of the number and variety of possible queries. On the one hand, the results of map queries often must be graphically represented, and their representation should be harmonious with that of the original map. Furthermore, if queries are made at a certain scale or involve only selected network elements, these properties and the ability to easily indicate the network element interactively must be maintained when constructing the map representation. If several, or complex, queries are required about discrete network elements, the types and degree of generalization appropriate may be more limited than if simpler results are desired. An engineer wishing to know the quality and location of all rails and switches in the London Underground would not be satisfied with the same map as a commuter desiring information about the shortest route between Westminster and Soho. Network map queries can be divided into two elementary kinds: *element* queries treat network components individually and can be answered without knowledge of the network as a whole, while *network*

queries involve the entire network and a knowledge of its organization to be answered correctly.

There are three types of element queries: simple, derived, and complex. These types may or may not involve distinctly spatial operations. None require a knowledge of network organization, and most can be answered on an element-by-element basis, making them candidates for parallel processing.

Simple attribute queries are answered directly by searching and retrieving information from a database. There are two methods of searching the database. Logical queries only require logical matching involving attributes, and the location of the elements involved is unimportant. These queries are essentially aspatial in structure, although they are often used to generate important spatial information. Identifying the location of all road segments with potholes can be a first step in organizing repairs or identifying possible reasons for their development. In locational queries, a particular network element is identified and its corresponding attributes are returned. These queries address relatively defined issues (essentially, they answer “what is here?” or “what is this?” questions). To create maps of these queries, a spatial join is performed on the generated data.

Derived attribute queries require some simple computer processing before they can be answered. The attribute requested is not directly stored in the database, but can be immediately derived from the stored data without knowledge of the entire network. Once again, these queries can take either logical or spatial forms. Logical queries of this sort will often involve simple case statements or greater than or less than quantitative evaluations. The example question, “where do traffic volumes exceed road design standards?” is quantitative. Spatially derived attribute queries are usually simple buffer and element-in-polygon queries (e.g., “show all roads within one mile of school X”): queries that can be answered with only a minimal knowledge of topology (e.g., contains, next-to) and do not require knowledge of the entire network. In many GISs, derived attribute queries produce new or intermediate coverages, while simple attribute queries do not.

Complex element queries are usually constructed from concatenated logical and spatial derived attribute queries. Theoretically, such queries could be entirely logical (for example, long and complex case statements involving attributes) or entirely spatial (less likely), but this is rarely the case. Most will involve some combination of spatial and aspatial characteristics (e.g., “show all road segments within 1.5 miles of a school that have traffic volumes that exceed road design standards”). Like derived attribute queries, complex element queries will usually result in the production of a new or intermediate coverage in a GIS.

Network queries cannot be resolved without an understanding of the entire functional network and its organization. Generally, they require longer and more complicated computer processing to arrive at a result. Many of the specialized data models and algorithms or techniques required by these queries are only now being incorporated into GIS or SDSS. Examples of such queries include finding the shortest path(s) between nodes, or finding the bottleneck in a network. As above, network queries can be concatenated, with the results of earlier network queries being necessary for other queries. For example, solving location-allocation models (e.g., “what is the optimal location for fire stations and allocation of districts to those fire stations?”) requires shortest-distance matrices, which are essentially constructed from many shortest path problems. The results

of the locational models, because they alter the structure or attributes of the network, may in turn require that earlier queries be recomputed. For example, relocating fire stations may change the locations of traffic bottlenecks. For purposes of this typology, all of these examples are classified as network queries.

CONCLUSION

Despite their importance to geography and geographic modeling, network representation and visualization strategies have not been well developed in the literature. This deficiency will become increasingly problematic as digital network-based modeling and analysis methods become more fully integrated into routine network maintenance and planning activities. On the one hand, to improve the automation of network visualization in an effective manner, some guidelines must be elucidated and codified. On the other, to maintain the quality (see Couclelis 1992) of maps delivered by GIS and related technologies, the increasingly nonspecialist mapmaker will require support in the application of basic cartographic principles. The integration of GIS and visualization techniques to support traditional and interactive location modeling is an emerging area of research (Allard and Hodgson 1987; Armstrong and Densham 1995a, 1995b; Armstrong et al. 1992; Densham and Armstrong 1995, 1993; Weibel and Buttenfield 1992).

Even if basic guidelines for network visualization techniques already existed, it is likely that they would need to be reevaluated at this time because network maps, and indeed mapping in general, has entered a new age of representation. In part, this transformation is driven by the general availability of a non-paper medium (the digital computer) with new resources for representation (animation, sound, hyperlinks) as well as more readily available digital data. Complicating this technical transformation are equally sweeping changes in the social demand for network mapping. Because digitally represented networks are being used in an increasing number of problem contexts by a diverse mix of users, digital systems not only must provide high-quality network maps in the variety of forms appropriate to these different contexts (which may or may not have cartographic typologies that are pre-existing and problem-specific) and often deliver these in near real-time, as well as provide personalized map types for individuals possessing different levels of cartographic skill.

This project shows how different data sources, with nominally unrelated spatial referencing methods, can be integrated into one data system that can be organized to answer spatially referenced questions of interest in transportation infrastructure management, transportation planning and spatial decision support for rural services planning. “Spatial referencing” is a relatively new organizing concept for data, and its usefulness in the field of transportation is not yet well known. Quite disparate information sources such as air photographs, census data files, census TIGER files, road management database files, and county digital cartographic files can, with suitable editing in specialized GIS software systems, be organized into one integrated data system. This project demonstrated for a five-county study region in Eastern Iowa, that existing data files produced by unrelated organizations, if shared, can produce one data system that is far more powerful for the purposes described above, than any of its parts.

Because an integrated geographic file that describes transportation infrastructure can be represented cartographically, a common misconception is that creating

and maintaining such a file is a cartographic function and users are those who, traditionally, would use cartographic products. This common view has stymied the development of such geographic files by limiting their use and reducing the number and variety of potential users and potential contributors. So many tasks can be performed more efficiently and effectively with access to an integrated geographic data infrastructure on transportation information, that developing and maintaining them should receive high priority.

REFERENCES

Allard, L., and M. J. Hodgson. 1987. “Interactive Graphics for Mapping Location-Allocation Solutions,” The American Cartographer, Vol. 14, No. 1, pp. 49–60.

American Cartographer, The. 1988. “The Proposed Standard For Digital Cartographic Data,” Vol. 15, No. 1, pp. 6–140.

American Hospital Association. 1993. *American Hospital Association Guide to the Health Care Field.* Chicago, IL.

Armstrong, M. P., and P. J. Densham. 1995a, “Cartographic Support for Collaborative Spatial Decision-making,” *Proceedings*, *Twelfth International Symposium on Computer-Assisted Cartography* (Auto-Carto 12), pp. 49–58.

Armstrong, M. P., and P. J. Densham. 1995b. “A Conceptual Framework for Improving Human Computer Interaction in Locational Decision-making.” In *Cognitive Aspects of Human-Computer Interaction for Geographical Information Systems.* T. L. Nyerges, D. M. Mark, R. Laurini and M. J. Egenhofer, eds. Dordrecht: Kluwer Academic Publishers, pp. 343–354.

Armstrong, M. P., P. J. Densham, P. Lolonis, and G. Rushton. 1992. “Cartographic Displays to Support Locational Decision-making,” *Cartography and Geographic Information Systems*, Vol. 19, No. 3, pp. 154–164.

Asproth, V., A. Håkansson, and P. Révay. 1994. “GIS Application for the Visualization of Streams,” *Computers, Environments, and Urban Systems*, Vol. 18, No. 2, pp. 103–109.

Astley, R. W. 1969. “A Note on the Requirements of Road Map Users,” *The Cartographic Journal*, Vol. 6, No. 2, p. 130.

Bertin, J. 1973. *Semiologie Graphique: Les Diagrammes—Les Réseaux—Les Cartes*. Paris, France: Gauthier-Villars.

Bertin, J. 1981. *Graphics and Graphics Information-Processing*. Translated by William J. Berg and Paul Scott. Berlin, NY: Walter de Gruyter.

Broome, F. R., and D. B. Meixler. 1990. “The TIGER Data Base Structure,” *Cartography and Geographic Information Systems*, Vol. 17, No. 1, pp. 39–47.

Buttenfield, B. P., and R. B. McMaster. 1991. *Map Generalization: Making Rules for Knowledge Representation*. New York, NY: John Wiley.

Clark, J. W. 1977. “Time-Distance Transformations of Transportation Networks,” *Geographical Analysis*, Vol. 9, pp. 195–205.

Clarke, K. C. 1995. *Analytical and Computer Cartography*. Englewood Cliffs, NJ: Prentice Hall.

Cromley, R. G. 1992. *Digital Cartography*. Englewood Cliffs, NJ: Prentice Hall.

Cutler, S. J., and R. T. Coward. 1992. “Availability of Personal Transportation in Households of Elders: Age, Gender, and Residence Differences,” *The Gerontologist*, Vol. 32, pp. 77–81.

Densham, P. J., and M. P. Armstrong. 1993. “Supporting Visual Interactive Locational Analysis Using Multiple Abstracted Topological Structures.” In *Proceedings of the Eleventh International Symposium on Computer-Assisted Cartography* (Auto-Carto 11). Bethesda, MD: American Congress on Surveying and Mapping, pp. 12–22.

Densham, P. J., and M. P. Armstrong. 1995. “Human-Computer Interaction: Considerations for Visual Interactive Locational Analysis.” In *Cognitive Aspects of Human-Computer Interaction for Geographical Information Systems*. T. L. Nyerges, D. M. Mark, R. Laurini and M. J. Egenhofer, eds. Dordrecht: Kluwer Academic Publishers, pp. 179–196.

Dent, B. D. 1996. *Cartography: Thematic Map Design*, 4th ed. Dubuque, IA: Wm. C. Brown.

Donabedian, A. 1983. *Aspects of Medical Care Administration*. Cambridge, MA: Harvard University Press.

Dorling, D. 1993. “From Computer Cartography to Spatial Visualization: A New Cartogram Algorithm,” *Auto-Carto*, Vol. 11, pp. 208–217.

Dueker, K. J. 1987. “Geographic Information Systems and Computer Mapping,” *Journal of the American Planning Association*, Vol. 53, No. 3 (Summer), pp. 383–390.

Dueker, K. J., and R. Vrana. 1992. “Dynamic Segmentation Revisited: A Milepoint Linear Data Model,” *Journal of the Urban and Regional Information Systems Association*, Vol. 4, No. 2, pp. 94–105.

Gould, P., and R. White. 1974. *Mental Maps*. Harmondsworth, Middlesex, England: Penguin Books.

Holtzheimer, P. E. 1983. “An Introduction to the GBF/DIME: A Primer,” *Computers, Environments and Urban Systems*, Vol. 8, No. 3, pp. 133–173.

Jenks, G. F. 1963. “Generalization in Statistical Mapping,” *Annals of the Association of American Geographers*, Vol. 53, No. 1, pp. 15–26.

Jenks, G. F. 1967. “The Data Model Concept in Statistical Mapping,” *International Yearbook of Cartography*, Vol. 7, pp. 186–190.

Joint Committee on Geographic Education. 1984. *Guidelines for Geographic Education*. Washington, DC: Association of American Geographers and National Council for Geographic Education.

Kadmon, N. 1982. “Cartograms and Topology,” *Cartographica*, Vol. 19, No. 3, pp. 1–17.

Kern, R., and G. Rushton. 1969. “A Computer Program for the Production of Flow Maps, Dot Maps, and Graduated Symbol Maps,” *The Cartographic Journal*, Vol. 6, No. 1, pp. 131–136.

Lew, A. A., and R. E. Klosterman. 1992. “Taking the TIGER by the Tail: A Software and Data Review,” *Journal of the Urban and Regional Information Systems Association*, Vol. 4, No. 2, pp. 106–117.

Lynch, K. 1960. *The Image of the City*. Cambridge, MA: The MIT Press.

MacEachren, A. M. 1986. “A Linear View of the World: Strip Maps as a Unique Form of Cartographic Representation,” *The American Cartographer*, Vol. 13, No. 1, pp. 7–25.

MacEachren, A. M. 1992. “Visualizing Uncertain Information.” *Cartographic Perspectives*, Vol. 13, pp. 10–19.

MacEachren, A. M. 1995. How Maps Work: Representation, Visualization and Design. New York, NY: The Guilford Press.

MacEachren, A. M., and D. W. DiBiase. 1991. “Animated Maps of Aggregate Data: Conceptual and Practical Problems.” *Cartography and Geographic Information Systems*, Vol. 18, No. 4, pp. 221–229.

McMaster, R. B., and K. S. Shea. 1992. *Generalization in Digital Cartography*. Washington, DC: Association of American Geographers.

Monmonier, M. 1991. *How to Lie with Maps*. Chicago, IL: University of Chicago Press.

Monmonier, M. 1993. *Mapping It Out: Expository Cartography for the Humanities and Social Sciences*. Chicago, IL: University of Chicago Press.

Morrison, A. 1971. “Experimental Maps of Road Travel Speed,” *The Cartography Journal*, Vol. 8, No. 2, pp. 115–132.

Nyerges, T. L. 1990. “Locational Referencing and Highway Segmentation in a Geographic Information System,” *ITE Journal*, March, pp. 27–31.

Robinson, A. H., and R. D. Sale. 1969. *Elements of Cartography*, 3rd ed. New York, NY: John Wiley.

Robinson, A. H., J. L. Morrison, P. C. Muehrcke, A. J. Kimerling, and S. C. Guptill. 1995. *Elements of Cartography*, 6th ed. New York, NY: John Wiley.

Rudnicki, R. 1992. “Assessment of TIGER/Line Files for Rural Counties: The Case of Hays County, Texas,” *Journal of the Urban and Regional Information Systems Association*, Vol. 4, No. 2, pp. 32–46.

Shenk, D. 1991. “Introduction: Serving the Needs of Rural Elders,” *Journal of Aging Studies,* Vol. 5, pp. 329–331.

Sheppard, D., and J. M. Adams. 1971. “A Survey of Drivers’ Opinions on Maps for Route Finding,” *The Cartographic Journal*, Vol. 8, No. 2, pp. 105–114.

Tobler, W. R. 1987. “Experiments in Migration Mapping By Computer.” *The American Cartographer*, Vol. 14, No. 2, pp. 155–163.

U.S. Bureau of the Census. 1993. *TIGER/Line Files, 1992*. Technical Documentation. Washington, DC.

Vujakovic, P. 1990. “Topological Transformations,” *Geographical Magazine*, Vol. LXII, No. 7, pp. 44–47.

Weibel, R., and B. P. Buttenfield. 1992. “Improvement of GIS Graphics for Analysis and Decision-Making,” *International Journal of Geographical Information Systems*, Vol. 6, No. 3, pp. 223–245.

Wittick, R. 1976. “A Computer System for Mapping and Analyzing Transportation Networks,” *Southeastern Geographer*, Vol. 16, No. 1, pp. 74–81.