To the Graduate Council:

I am submitting herewith a thesis written by Todd Christopher Raynor entitled "Porting Transportation Models Across GIS Platforms: A Case Study of TransCAD, Arc/Info, and The Highway Traffic Forecasting System." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Geography.

Bruce A. Ralston, PhD, Major Professor

We have read this thesis
and recommend its acceptance:

Thomas L Bell

Cheng Liu

Accepted for the Council:

Associate Vice Chancellor
and Dean of The Graduate School

PORTING TRANSPORTATION MODELS ACROSS GIS PLATFORMS: A CASE STUDY OF TRANSCAD, ARC/INFO, AND THE HIGHWAY TRAFFIC FORECASTING SYSTEM

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Todd C. Raynor

August, 1995

ACKNOWLEDGMENTS

I am deeply indebted to my wife, Hollie, for her encouragement, support, and belief in my abilities to succeed in college. Without her, this thesis would never have been completed.

I am also indebted to Dr. Fredrick Stutz who sparked my interest in GIS-T and encouraged me to pursue graduate school. To Dr. Bruce Ralston, I extend my deepest appreciation and admiration. Seldom is one so lucky to work with a man of such intellect and, possibly more important, humor. I will greatly miss his guidance. In addition, I extend my thanks to Drs. Thomas Bell and Cheng Liu for their comments and help in completing this thesis.

Finally, I send my gratitude to Dr. Julian Ray for his help and guidance throughout my graduate work, and particularly on this thesis. Working on a daily basis with Dr. Ray has given me an entirely new perspective on approaching and solving problems. It seems every conversation with Dr. Ray is an educational experience, and I hope we will continue to work closely in the future.

ABSTRACT

Geographic Information Systems (GISs) are excellent platforms for managing and analyzing transportation data. However, there are a tremendous number of different problem-specific transportation models being used today, and they often require specific data structures and data types which commercial GISs do not readily support. It is not feasible for commercial GISs to include all the analytical models being used for transportation analysis; therefore, it is often necessary for users to integrate their own problem-specific models within GISs.

This thesis examines some major issues regarding transportation analysis within GISs, and develops an approach for porting transportation models across GIS platforms and integrating them within GISs. In particular, this thesis shows how the Highway Traffic Forecasting System (HTFS) was ported from TransCAD, a DOS based GIS, and integrated with Arc/Info GIS on a Unix workstation. The study presents the procedures used to port HTFS, and explains the software routines developed to manage the data flow and create the data structures required by HTFS. In addition, the development of a comprehensive graphical user interface, and HTFS's use within Arc/Info, is presented.

TABLE OF CONTENTS

CHAPTER PAGE

1. INTRODUCTION AND REVIEW 1

    REVIEW OF GEOGRAPHIC INFORMATION SYSTEMS 2

    TRANSPORTATION ANALYSIS IN GIS 8

    INTEGRATION OF HTFS AND ARC/INFO 18

2. INTRODUCTION TO THE HIGHWAY TRAFFIC FORECASTING SYSTEM 21

    OVERVIEW OF HTFS 21

    HISTORY OF HTFS 22

    HTFS AND TRANSCAD 24

    MODELS AND ROUTINES 28

    HTFS DATA TABLES 39

3. INTRODUCTION TO ARC/INFO, NETSHELL, AND TED 44

    ARC/INFO MODULES 45

    SPATIAL FEATURES AND TOPOLOGY 50

    NETSHELL AND TED 53

4. INTEGRATION OF HTFS AND ARC/INFO 61

    PROCEDURES FOR INTEGRATION 61

    DESIGN AND DEVELOPMENT OF AML CODE 75

5. IMPLEMENTATION OF HTFS 89

    THE BASECASE SCENARIO 89

SCENARIO 1: Allowing Triples on All Interstates 91

SCENARIO 2: A New Interstate Between Denver and Phoenix 96

SCENARIO 3: Increasing the Fuel Tax by 10 Percent 106

6. CONCLUSION AND DIRECTIONS FOR FURTHER RESEARCH 113

BIBLIOGRAPHY 119

APPENDICES 127

Appendix A 128

Appendix B 133

Appendix C 146

Appendix D 171

Appendix E 182

Appendix F 192

Appendix G 193

VITA 197

LIST OF FIGURES

FIGURE PAGE

1-1 Relational Data Model 5

1-2 Example of NHPN Version 2.0 Link File 10

1-3 Example of a Link File Displayed in a GIS 11

2-1 TransCAD's Relationship with HTFS 25

2-2 TransCAD's Interface and Procedure Selection Menu 26

2-3 The Relationship Between HTFS's Main Routines and Input/Output Files 29

2-4 Iterative Proportional Fitting Routine 38

3-1 Arc/Info's Georelational Linkage for a BEA Coverage 51

3-2 Arc/Info Relation Between a FAT and a Data Table 52

3-3 Arc and Node Topology 54

3-4 NETSHELL's GUI 56

3-5 TED's GUI 57

4-1 Example of Arc/Info's CLEAN command 65

4-2 Example of External Polygons in California 66

4-3 Example of TransCAD Data Translation 68

4-4 Flowchart of HTFS's GUI 77

4-5 AML Widgets 79

4-6 How HTFS.AML Interprets HTFS.MENU Selections and Executes Routines. 81

4-7 Example of a Scenario File Written by FNET.AML 84

4-8 Flow of HTFS's Freight Network Policy Model Option, and When Information is Written to ASCII files..... 86

5-1 Basecase Flows of Triples on Interstates..... 93

5-2 Forecasted Flows of Triples on Interstates..... 94

5-3 Triples Allowed on All Interstates - National VMT and PTM Totals..... 97

5-4 Location of New Interstate..... 99

5-5 Increase or Decrease in VMT after New Interstate is Added..... 102

5-6 Flow Diagram of Basecase VMT without Interstate..... 103

5-7 Flow Diagram of Forecasted VMT with New Interstate..... 104

5-8 New Interstate Between Denver and Phoenix - VMT and PTM Totals for Arizona, Colorado, New Mexico, and Utah..... 105

5-9 New Interstate Between Denver and Phoenix - National VMT and PTM Totals..... 107

5-10 Percent Change in VMT after 10 Percent Fuel Tax..... 109

5-11 Percent Decrease in VMT after 10 Percent Fuel Tax..... 110

5-12 Ten Percent Increase in Fuel Tax - National VMT and PTM Totals..... 112

CHAPTER 1

INTRODUCTION AND REVIEW

Geographic Information Systems (GISs) are used for management, analysis, and display of spatial data. The types and uses of spatial data are diverse and varied, as are GISs. The diversity of GISs is seen in the different problems they solve. They are being used by businesses to locate and target clients; by health professionals to track and monitor the diffusion of diseases; by the forest service to manage the harvest of timber; and by transportation professionals to store, predict, and manipulate data on transportation networks. This diversity has caused GIS vendors to utilize many different approaches for developing GIS software. Some vendors develop their GIS for specific niches in the market while other vendors have developed more generic GIS software to solve many different and divergent problems.

GISs are excellent platforms for modeling transportation because of their powerful display and analysis capabilities; however, few commercial vendors have developed GISs specifically for exploring transportation issues. Furthermore, there are many non-GIS based transportation software packages which could benefit from the strengths found in commercial GISs. Consequently, it is becoming increasingly common for transportation professionals to integrate their own models within GISs.

This thesis develops one approach for integrating transportation models within a GIS. The study will show how the Highway Traffic Forecasting System (HTFS), developed jointly by the University of Tennessee's Transportation Center and the Oak Ridge National Laboratory, was integrated with a GIS running on a Unix workstation.

HTFS is a modeling system designed to model truck freight movements at a national level, and was originally designed to run under Caliper Corporation's TransCAD GIS. This thesis describes the procedures used to port and integrate HTFS with ESRI's Arc/Info GIS software.

This chapter presents an overview of Geographic Information Systems, followed by a discussion of transportation modeling issues related to GIS. It closes with an explanation of the purpose and procedures of this study. The subsequent chapters give a comprehensive description of HTFS, Arc/Info, the procedures used for integration, and a demonstration of HTFS's use within Arc/Info. The final chapter summarizes the results, and gives directions for further research.

1.1 REVIEW OF GEOGRAPHIC INFORMATION SYSTEMS

The definition of what constitutes a GIS has been intensely debated in current literature (Maguire 1991). GISs have created tremendous interest throughout the world, because they are capable of helping users understand and analyze vast problems in disparate fields. There is a tremendous market for GISs, and, in response, a tremendous growth of computer software purporting to be a GIS in an attempt to exploit that market (Maquire 1991). The growth of software purporting to be a GIS, many of which simply store and display spatial data, has added to the complexity of adequately defining what a GIS is or should be. However, when a GIS is defined, there is one common feature; GISs deal with spatial data (Maquire 1991). The National Center for Geographic Information and Analysis (NCGIA) defines a GIS as "a computerized database management system

for capture, storage, retrieval, analysis, and display of spatial (i.e. locationally defined) data" (Simkowitz 1990a, Zhu 1992), and this definition has become accepted and common throughout much of the current literature.

The key feature of a GIS is its ability to perform operations on spatial data. Through the use of a database management system (DBMS) for storing aspatial (i.e., attribute) data with internal linkages referencing spatial (i.e., locational) data, a GIS is able to perform many different statistical, analytical, and querying operations within a single geographic data table or between diverse sets of data which have been spatially referenced and geocoded to a standard referencing system (Simkowitz 1990b). In order to manage and execute these operations, GISs are usually composed of several different and distinct modules, typically a DBMS, graphical display routines, and spatial analysis algorithms.

1.1.1 Database Management Systems

DBMSs are typically highly developed software programs that store, retrieve, and manipulate non-graphic aspatial data (Maguire 1991). A DBMS can be exploited by a GIS for retrieving data from a data table via simple or complex selection routines and then feeding the data to the GIS's analytical routines for further processing (Healey 1991). Although there are many different DBMS models from which to choose, the relational data model is the type most widely used by GISs for storing and managing aspatial data. The relational model has several characteristics which make it a preferred choice for GISs. The relational model is developed so that data can be normalized (i.e.,

non-duplicated), and the relational model is typically easy to use and flexible (Healey 1991). A relational model allows data to be normalized through the use of tables which can be joined by using relational algebra. The data are distributed in different tables so that duplication of data is minimized, and tables can be joined together based on matching items. Figure 1-1 shows three different tables and how they are related. The method used to link different tables to each other is called a relational join. In a relational join, values from one or more columns in one table (the primary key) are matched to values in another table's column(s) (the foreign key), whose column(s) values can be further matched to a third table's column(s), and so on (Healey 1991).

1.1.2 Data Models for Graphical Display and Feature Representation

One deficiency of a relational data model is its inability to store locational coordinates so they can be accessed and displayed in an efficient manner. Spatial features such as lines and polygons can have hundreds or thousands of coordinates associated with them. These coordinates need to be accessed by a GIS's display routines so the features can be graphically displayed and analyzed. If the coordinates are stored in a DBMS, access to them is inefficient because they have to be retrieved from the DBMS. Many GISs have, therefore, developed a hybrid of the relational data model where the coordinates of features are accessed directly from operating system files, greatly increasing the speed of data input and output (Healey 1991). Arc/Info is an example of a GIS which uses this hybrid data model.

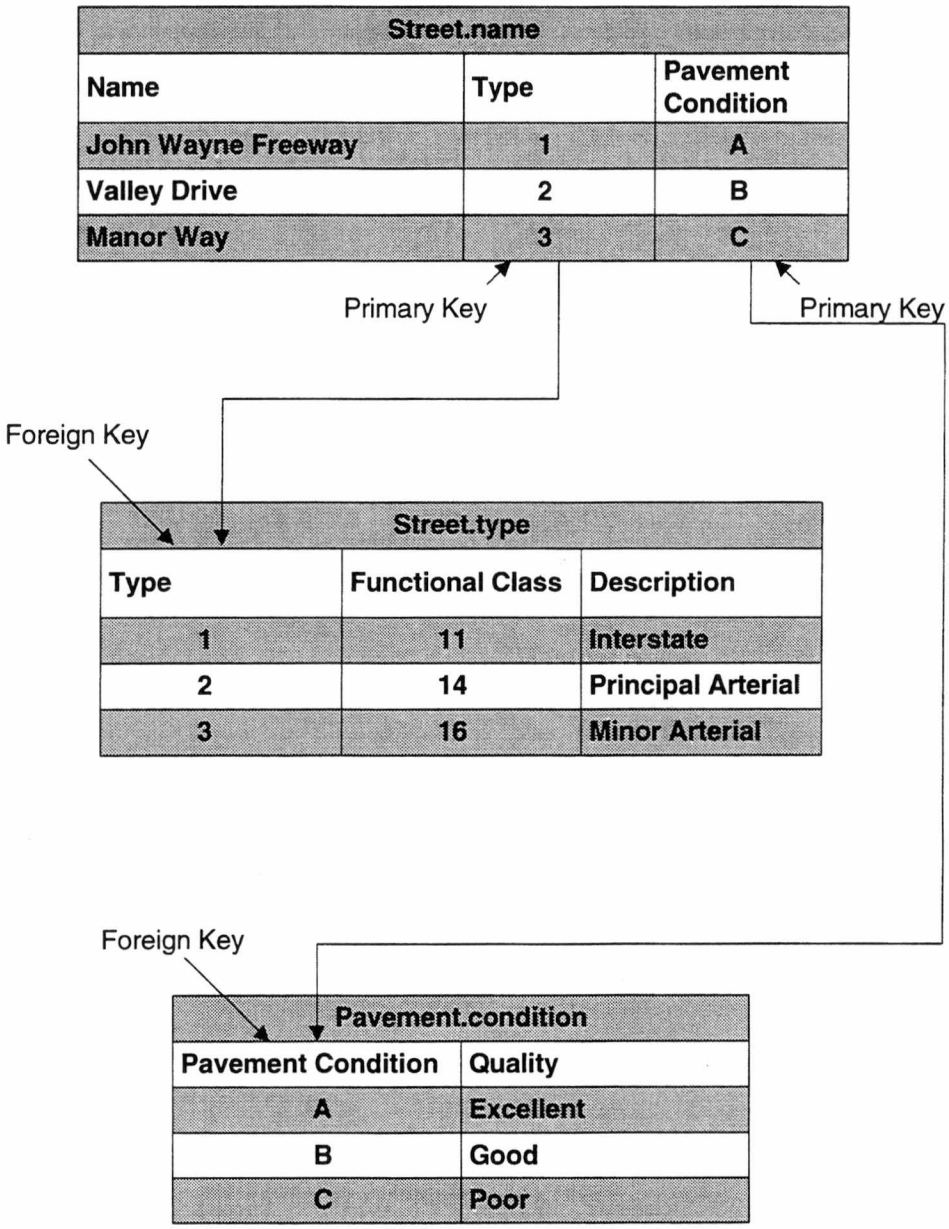


Figure 1-1
Relational Data Model

GISs use two common methods for storing and displaying spatial data: raster and vector. Raster data represent reality (features occurring in space) by a rectangular matrix of cells. Each individual cell has a value representing a geographic phenomenon. Vector data represent reality by depicting two-dimensional maps as points, lines or polygons on an (x,y) Cartesian coordinate system. These points are used to reference map locations to ground locations. A point is represented by an (x,y) location, a line is represented by a series of (x,y) locations which define its shape, and polygons are represented by a series of (x,y) locations (typically lines) beginning and ending at the same point, thus enclosing an area. With vector data, points, lines, and polygons can be represented as a list of (x,y) coordinates instead of being represented as a graph or picture (ESRI 1994). Many GISs utilize vector data because it is more precise than raster; vector data use exact locations instead of areas (i.e., cells), thus representing shapes more accurately.

Another advantage vector data has is topology, a method for defining relationships between spatial features. Topology has three major concepts; lines connect to each other at nodes (the endpoints of lines), connected lines surrounding an area define a polygon, and lines have direction and right and left sides. By using topology, data can be processed much faster, analytical functions, such as modeling flows through a network, are facilitated, and identifying adjacent features is relatively simple (ESRI 1991). Thus, many have argued that topology is one of the key concepts in GISs.

Typically, GISs represent topology through internal linkages or pointers. A line's attribute table (which stores the non-locational attributes of spatial features) will normally have columns representing topology. These items consist of a *from-node*, a *to-node*, a

left-polygon, and a *right-polygon*. Since the arc attribute tables contain information about the beginning and terminating nodes of each arc, the arcs can be chained, thus enabling analysis routines, such as shortest path algorithms, to be performed on the data. If the spatial features contain polygon topology, then the *left-polygon* and *right-polygon* fields of the arc attribute table will reflect this information. Contiguous polygons will share a common boundary, or arc. Thus, polygon features adjacent to one another can be identified and analyzed.

Data organization is another fundamental concept in GISs. Data are typically organized into themes, layers, or coverages (synonymous terms depending on the GIS) so that different types of features can be maintained separately. The separation of data is typically made based upon feature type (point, line, or polygon), or thematically (streams, roads, or waterways). By storing data based on feature type or thematic attributes, the data set is much more manageable.

1.1.3 Spatial Analysis

A GIS is typically composed of a wide range of analysis routines capable of acting on topology or spatial aspects, aspatial attributes, or spatial and aspatial attributes combined. Analysis routines vary from simple methods for retrieving subsets of data, to spatial analysis such as using neighborhood functions and interpolation (Burrough 1986). For example, depending on the GIS software, analysis routines can be used to select spatial features encompassed by a polygon in another layer, find the intersection of features between different layers, or create a union of features between layers.

Some (Openshaw 1991) have argued that many of the operations described above are not really spatial analysis routines, except in a cartographic or data descriptive sense. Openshaw (1991) says that "spatial analysis was originally based on the application of the available statistical methods to spatial data.... Later, it was extended to include mathematical model building, and operational research methods." The lack of true spatial analysis routines available in many of the commercial GIS packages has been criticized in much of the literature (Densham 1991, Gatrell 1991, Maquire 1991, Lewis 1990). Therefore, the areas of applied quantitative, statistical, and mathematical analysis and modeling need to be integrated into GISs in order to increase their functionality and usefulness (Openshaw 1991). This is especially true for transportation problems which require complex mathematical models to be solved.

1.2 TRANSPORTATION ANALYSIS IN GIS

Transportation is inherently a spatial problem and, as such, the transportation field has a great interest in using GIS for storing, managing, modeling, analyzing, and displaying their data. Developing GISs to specifically analyze transportation issues, often called GIS-T, is a fairly recent phenomenon, extending back only to the late 1980s. The field of GIS-T has been spurred by the Federal Government's Intermodal Surface Transportation Efficiency Act of 1991 (ISTEA) which allocated a tremendous amount of funding to improve the nation's transportation infrastructure. According to Ray and Armstrong (1994) ISTEA, and the funding accompanying it, has helped to define and

develop the field of GIS-T, resulting in the knowledge of where GIS excels and what GIS lacks or performs inefficiently for solving transportation issues.

Models for predicting flows of people or goods from an origin to a destination, projecting the number of cars on a network, and determining what mode of transportation a commodity will choose are extremely data intensive. Large amounts of data describing the transportation network, attributes of the user, commodity, or mode, and characteristics of origins and destinations are required to model transportation flows. Transportation professionals use computers to process these large data sets. The decreasing cost and increasing speed of computers have dramatically aided the transportation sector in its ability to efficiently model transportation. GISs are further aiding the field because of their powerful data management, editing, analysis, and display routines which can handle large databases, such as those used for transportation analysis.

Prior to the development of a GIS-T, transportation data were often stored in non-spatial files. These data files were read by statistical packages or models, and generated non-spatial result files. The compilation and interpretation of these files was time consuming, error prone, and laborious. For example, Figure 1-2 shows a sample of a link file from the National Highway Planning Network Version 2.0 (NHPN 1994). Each line in this figure represents a single link and its attributes. By viewing this file, it is apparent that managing the attributes and topological relationships of each link would be quite complex. GISs allow transportation data to be displayed, edited, and analyzed from a graphic screen. Figure 1-3 shows an example of how a network's attributes can be edited through a menu. Utilizing a GIS's powerful graphical display allows transportation data

```
L20110000031100111000118008000001100000711000010U29          GA-AV      3 420    432114P31700
L20110000171100111000117008000001100001011000022U29          GA-AV    175 28220 432114P31700
L20110000321100111000326008000001100003811000026U1           RI-AV    223 35920 431114P31000
L20110000341100111000042008000001100003611000032U50          NY-AV     95 15320 421114P31700
L20110000351100111000115008000001100002211000039U29          GA-AV    139 22420 432114P31700
L20110000371100111000324008000001100004011000038U1           RI-AV     46 7420  431114P31000
L20110000401100111000113008000001100003911000041U29          7-ST      28 4620  432114P31700
L20110000451100111001030008000001100004511000043U29          RI-AV     34 5520  431114P31000
L20110000481100111000043008000001100004711000036U50          NY-AV    183 29420432114P31700
L20110000501100111000044008000001100004811000047U50          NY-AV     26 4120 432114P31700
L20110000581100111000048008000001100005311000048U50          NY-AV     39 6220 432114P31700
L20110000591100111001000008000001100004511000054U29          13-ST     34 5520 432100P31000
L20110000651100111000240008000001100005711000053U50          NY-AV     36 5820 432114P31700
L20110000701100111000234008000001100006011000059U29          K-ST      24 3920 431114P31700
L20110000721100111000233008000001100005811000060U29          K-ST      15 2420 431114P31700
L20110000731100111000860008000001100005411000061U29          13-ST     17 2720 432100P31000
L20110000741100111000235008000001100005911000061U29          K-ST      13 2120 431114P31000
L20110000771100111000232008000001100006311000058U29          K-ST      54 8820 431114P31700
L20110000801100111000680008000001100005211000062U29          WHITE     64 10320411100P31700
L20110000811100111000231008000001100006211000063U29          K-ST      45 7320 431114P31700
L20110001201100111000208008000001100008911000090U50          CONST      9 1520 432114P31700
L20110001211100111000201008000001100008411000088U50          CONST     19 3020 432114P31700
L20110001221100111000206008000001100008611000089U50          CONST     31 4920 432114P31700
L20110001231100111000203008000001100008811000091U50          CONST     12 1920 432114P31700
L20110001241100111000205008000001100009111000086U50          CONST     45 7220 432114P31700
L20110001251100111000211008000001100009011000087U1  U50      CONST     22 3520 432114P31700
L20110001271100111000212008000001100008711000092U1  U50      CONST     21 3420 432114P31700
L20110001301100111000218008000001100009211000094U1  U50      CONST     12 1920 432114P31700
L20110001451100111000258008000001100009011000107U1           14-ST     31 5020 432114P31700
L20110002261100111000300008000001151024211240243I95                    13 2120 411111I31100
```

Figure 1-2: Example of NHPN Version 2.0 Link File

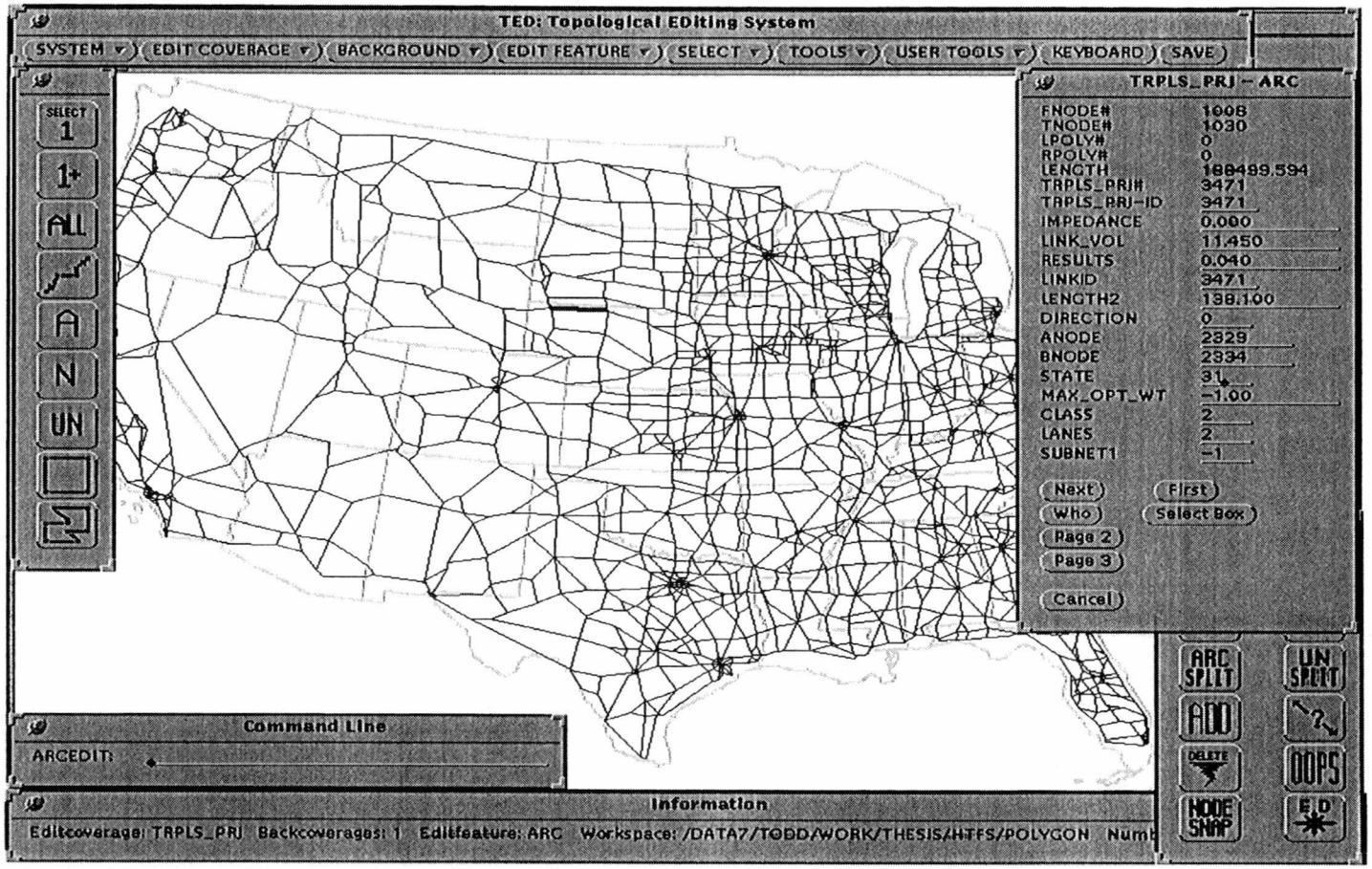


Figure 1-3: Example of a Link File Displayed in a GIS

to be dealt with in an efficient and effective manner. For example, data tables can be managed through a GIS's graphical display by querying features and editing attributes with a mouse.

Topological editing can also be performed with a mouse. A feature can be added, deleted, repositioned, reshaped, etc., via a mouse, and the data table will immediately reflect that change. In addition, features and results of analysis can be displayed and queried effectively by using a wide range of colors and symbology. GISs are therefore a tremendous platform for transportation professionals to use in their work, and GISs are being applied to many different transportation applications. For example, they are being used for pavement, bridge, and safety management; travel demand forecasting; hazardous cargo routing; and accident and environmental impact analysis (Petzold and Freund 1990). HTFS also uses a GIS to aid in modeling freight movements on the nation's highways.

Although being widely adopted for use in the transportation sector, GISs still lack many of the features and functionalities transportation problems demand. A more in-depth look into GIS and its functionality related to transportation reveals areas where it is not yet adequate. Ray and Armstrong (1994) define several areas where GIS related to transportation is lacking. They include the following areas: data structures, editing tools, visualization tools, data standardization, and algorithm development.

Network Data Structures

Most data structures used by GISs today were developed to operate on environmental problems, typically consisting of polygon data, and current GISs are normally not well developed for handling linear network data (Lewis 1990). Transportation models require specific data structures that are quite different from conventional spatial data structures. This is because, among other reasons, many attributes on each link on a transportation network may need to be evaluated to determine a particular link's characteristics, and many different data types (polygon, point, and lines) are often required for model input. In addition, transportation models' data requirements include data files containing matrices, vectors of data, and topology of networks containing link and node attributes (Simkowitz 1990a). Therefore, GISs should be able to handle many different data structures in order to model transportation effectively.

Network data tables can be extremely large; thus, data structures which are computationally efficient need to be integrated into GIS-Ts (Simkowitz 1990a, Ray and Armstrong 1994). An example of a computationally efficient data structure is a forward star network which stores the connections between links and nodes. The connections are stored by referencing the internal identifier's of the link, the node it leaves, and the node it enters. By storing link information in this manner, links can be quickly chained, and the pertinent attributes can be referenced by pointers to attributes in data tables (Ray and Armstrong 1994).

Network Editing

Attributes and topology of networks are extremely dynamic. Roads are constantly being resurfaced, widened, realigned, and built. A number of network editing tools should be included with GISs so changes to the attributes or geometry of a road can be easily incorporated. This is especially true when complicated network data structures are used, such as those found when storing linear referencing schemes (LRS) (Ray and Armstrong 1994). A LRS, which is defined by Okunieff et al. (1994) as "a method for determining the position of an entity relative to other entities or to some external frame of reference," requires extremely complex data structures to display multiple representations of a road network (Lewis 1990). The complex nature of recording and storing this information requires the development of tools to aid the user in managing these types of data. The Route Editing System (REDS 1995), designed by members of the GIS Group at the University of Tennessee's Transportation Center, is an example of one network editing system which is designed specifically for building and maintaining linear referencing schemes.

Network Visualization

Due to a network's vector representation, links are assumed to have a single spatial dimension; however, in reality, these features may have two or even three dimensions, such as length, width, and grade. Displaying multiple attributes on a non-areal feature impose problems. By using color and line widths, two attributes can be displayed effectively (e.g., road class by color and traffic volume by line thickness), but

when displaying more than two attributes, the level of detail which can be depicted is quickly diminished (Ray and Armstrong 1994). In addition, networks often contain bi-directional data. Bi-directional data such as link flows or number of lanes is extremely difficult, if not impossible, to display simultaneously in most GISs. Another deficiency is the representation of multi-modal networks. Multi-modal networks typically use exploded nodes to represent transfer points between different modes of transportation. Exploded nodes contain logical links and nodes to represent different modes of transportation. They do not actually exist in the network's data table. Thus, this information cannot be displayed in the traditional GIS environment (Ray and Armstrong 1994).

Data Standardization

Data standardization is another concern of the GIS-T community as well as other GIS users. The Spatial Data Transfer Standards (SDTS) serve as a mechanism for the transfer of national spatial databases for all federal agencies, and are available for use by state and local governments, as well as the private sector (Hogan 1993). As yet, however, SDTS has not been developed for transportation data. Thus, methods for storing entities such as paths and tours or linear referencing schemes have not yet been addressed (Ray and Armstrong 1994). The adoption of common data standards, allowing data to be transferred more readily between or within agencies is, therefore, of particular importance to GIS-T (Lewis 1990).

Algorithm Development

As discussed earlier, most GISs lack the numerous modeling routines, or adequate models, required by the transportation sector. Simkowitz (1990a) stresses the importance of integrating a large number of transportation application modules capable of handling extremely large networks. These large networks often pose a problem in that they may be too large for direct input into transportation models. Therefore, there is a need for new algorithms to be developed which could intelligently generalize a network into a condensed form, and, in turn, pass the condensed data to the transportation models (Ray and Armstrong 1994, Ralston et al. 1994, Ralston 1989).

1.2.1 Strategies for Integrating GIS and Transportation Models

The large number of models needed for transportation analysis impose great demands on GISs. In addition, there are typically numerous different model formulations to solve the same problem; thus, many institutions possess their own specific models which have no bearing on a GIS's capabilities (Chen and Tseng 1994). Since there are so many different models being used by the transportation community, it is impossible for GISs to include all of them.

One attempt to bridge the gap between traditional GISs and GIS-T is described by Simkowitz (1990a). Simkowitz explains how Caliper Corporation's TransCAD GIS has developed a hybrid architecture incorporating significant transportation data structures and transportation specific procedural modules for data input, processing, and output. Caliper has developed a suite of generic transportation models commonly used in

transportation analysis; thus, the most common transportation issues can be evaluated with their off-the-shelf software. For transportation issues requiring specialized models, TransCAD allows models developed outside their software domain to be easily accessed through their user interface. This allows users to take advantage of TransCAD's DBMS, graphical display capabilities, and spatial analysis routines, send their data to their own problem-specific models for processing, and import the results back into TransCAD for analysis. This approach is promising for GIS-T because the most common transportation models are included with the software, but the ability to access specialized models developed by the user, or other transportation professionals, can be easily accessed through the GIS's menu.

Others have developed GIS-T systems for solving specific problems. These are highly developed systems which include all of the models, data structures, and graphical display routines required to define, model, and analyze a particular problem. An example of these systems are the Bangladesh Transportation Modeling System (Liu and Ralston 1990), and the Transportation Inland Logistics Manager (Ralston et al. 1992). Although these systems are limited in their functionality for analyzing diverse issues, they supersede the compatibility problems normally found when integrating transportation models with GISs (Zhu 1992).

A third approach is to create tools for building transportation model structures from GIS databases. Ralston (1994) has argued that object-oriented programming (OOP) approaches offer the best strategy for building such tools. Zhang (1993) has taken that

idea further by developing an OOP-based program generator for building matrix models from GIS databases.

The Highway Traffic Forecasting System bridges the first two approaches described above. HTFS was designed to read the most common ASCII file formats (i.e., comma delimited and columnar) exported by GISs, and reformat these data into structures optimized for its models. Although the initial data processing is inefficient because it is necessary to export the data from the GIS to ASCII files and then convert them to an optimal format, this approach allows HTFS to be GIS-independent and portable. Thus, HTFS can take advantage of a GIS's functionality without being tied to, or limited by, the GIS's data model.

1.3 INTEGRATION OF HTFS AND ARC/INFO

Aside from TransCAD, few commercial GISs have been developed which are dedicated to, or easily allow, transportation analysis. Since the trend for GISs has been "towards workstations running the Unix operating system" (Maguire 1990), and most transportation models have been developed to run on a DOS platform (Lewis and McNeil 1986), the integration of transportation models into fully functional Unix based GISs is further complicated. The Highway Traffic Forecasting System was written so it can be compiled and run on either a DOS or Unix platform. Originally developed to run under TransCAD, HTFS was designed so it could be ported and integrated with a GIS running on a Unix platform. Although HTFS has been used by the Federal Highway Administration (FHWA) to solve policy issues regarding freight transportation under

TransCAD, it has never been integrated and tested with a GIS running on a Unix platform.

1.3.1 Purpose of Porting and Integration

GISs running on a Unix platform offer advantages over those running on a microcomputer DOS platform. Typically, workstations operate at faster speeds and have more RAM and disk storage than DOS-based PCs. However, these differences are beginning to become blurred. Because transportation models are memory-intensive and can take an inordinate amount of time to solve, a workstation's typical large memory configuration and high CPU speeds make them a preferable platform for performing spatial analysis and running transportation models.

Porting HTFS to a Unix platform allows it to be integrated with Arc/Info GIS which contains a much wider range of spatial analysis routines, greater display capabilities, more robust DBMS functions, and a greater selection of editing tools then TransCAD offers. Therefore, by porting HTFS to a Unix workstation and integrating it with Arc/Info, HTFS can process data faster, process larger models, and take advantage of a much wider range of GIS tools, thereby increasing its efficiency.

1.3.2 Procedures for Porting and Integrating HTFS

In order to integrate HTFS within Arc/Info, several steps will be required. The steps include:

- Developing an understanding of HTFS's original usage and design within TransCAD.
- Understanding the data structures of TransCAD and Arc/Info, and writing translation routines so HTFS's data can be imported into Arc/Info.
- Compiling HTFS's models on the workstation.
- Studying the data flow and data formats of HTFS's routines and writing routines in Arc/Info's Macro Language (AML) to control the data flow and reproduce the required data formats.
- Modifying HTFS's routines to output data formats suitable for import into Arc/Info.
- Developing a comprehensive, easy-to-use, Graphical User Interface (GUI) in AML to aid the user in scenario definition, running the models, and importing the results.
- Developing scenarios and testing the integration.

These steps will be discussed in detail in Chapter 4, but first an overview of HTFS and Arc/Info will be given in Chapter's 2 and 3, respectively, to give the reader an introduction to these systems.

CHAPTER 2

INTRODUCTION TO THE HIGHWAY TRAFFIC FORECASTING SYSTEM

Sound transportation planning has a strong impact on the national economy by affecting the speed, cost, and efficiency of transporting goods throughout the country. By introducing governmental policies such as user taxes or weight restrictions on the nation's roadways, the cost of goods at the market can be dramatically affected. There are numerous parameters involved in attempting to optimize the nation's road network and the policy decisions accompanying it, and many different transportation models have been designed to analyze and forecast different network design and policy scenarios. The Highway Traffic Forecasting System is one such model. This chapter will focus on the history of HTFS, the current relationship HTFS has with TransCAD, its primary models and routines, and the data tables used by HTFS.

2.1 OVERVIEW OF HTFS

The Highway Traffic Forecasting System (HTFS) is an assemblage of transportation models and data tables used for analyzing a wide range of national highway planning policy issues. The primary purpose in the development of HTFS was to integrate data on truck activity and to produce policy-sensitive predictions of national vehicle miles of travel (VMT) and payload-ton miles (PMT) and how they are distributed across different truck types and operating gross weight (OGW) intervals. HTFS was designed for making 5-10 year forecasts on freight movements, vehicle activities, and highway use. The forecasts made by HTFS are highly dependent on policy issues such as

weight restrictions, vehicle size, highway user taxes, as well as demographic and economic trends at the regional or national level. HTFS was designed for analyzing policy scenarios which affect truck freight, and the system includes the parameters and data necessary to analyze many different governmental policies which may be proposed (Chin et al. 1992). HTFS was designed for use by the FHWA's Office of Policy Development, and has been in existence for nearly a decade in several different forms.

2.2 HISTORY OF HTFS

Originally, HTFS was developed as a series of Statistical Analysis System (SAS) programs and FORTRAN routines. It was initially developed by the Department of Transportation's Transportation Systems Center in Cambridge, Massachusetts in 1985. The first documentation for the system became available in 1986 (Neinhaus 1986), and staff at Oak Ridge National Laboratory became involved in reviewing the system shortly thereafter (Southworth and Peterson 1986). The system was originally designed to be run on a mainframe computer, but managing the system proved to be too complicated and cumbersome because of the numerous data files required, order of processing, and its arcane interface. To make the system easier to use, HTFS began evolving towards a system containing a more user-friendly interface. Chin et al. (1992) describes how:

> with the continued advances in microcomputing and in particular in the development and desire for network-based GIS tools, it was decided to make at least the spatially sensitive components of HTFS operational in a modern day GIS environment.

The evolution of HTFS to a GIS-based transportation analysis package resulted in the development of the Freight Network Policy Model (FNET) by Ray and Southworth

(1992) in cooperation between the University of Tennessee's Transportation Center and Oak Ridge National Laboratory's Center for Transportation Analysis.

FNET has become the major component of HTFS and runs on a DOS PC under TransCAD Version 2.0 GIS software. FNET is implicitly a spatial model designed with network optimization and transportation planning methodology, and was developed to address scenarios with a major spatial component. As such, FNET is the primary modeling component of HTFS interfacing with TransCAD (Ray 1992). FNET is designed to appraise changes in annual VMT and PTM by truck class due to changes in Truck Size and Weight (TS&W) limits and policy issues. The forecasts of FNET can then be further analyzed by other non-spatial policy models in HTFS. Chin et al. (1992) lists five additional policy models in HTFS which FNET's results can be input into:

1. Revenues to the Highway Trust Fund by type of vehicle and highway user group,
2. Pavement and bridge damage and other infrastructure costs,
3. The system cost responsibility of each class of highway user,
4. The impact of highway policies on other modes of transportation, especially rail, [and]
5. Changes in transportation costs incurred by highway users attributable to changing patterns of highway use.

The remainder of this chapter will discuss TransCAD, and the models, files, and spatial analysis features of HTFS, also called FNET.

2.3 HTFS AND TRANSCAD

HTFS has numerous input and output files associated with a scenario, and the management of these files can be extremely complex. Finding an appropriate and somewhat intuitive method for managing these files was necessary to increase the usability of HTFS, and TransCAD provides many of the tools necessary to accomplish this. TransCAD's database manager, display capabilities, and the ability to integrate user routines, for example, allowed HTFS to become a manageable modeling system. Figure 2-1 demonstrates the basic relationship between TransCAD and HTFS when used for policy analysis.

TransCAD is a vector-based menu-driven GIS developed for "planning, management, operation, and analysis of transportation systems and facilities" (Caliper Corporation 1990). One of the primary features of TransCAD is the ability to integrate external software, such as HTFS's procedures, into it with minimal effort. A procedure is a DOS program developed to perform an operation on a TransCAD data table. Most of HTFS's procedures are written in the C programming language, although some are written in FORTRAN. With TransCAD, files can be constructed, similar to DOS batch files, which allow the user to issue commands to TransCAD and interface with HTFS's procedures. TransCAD's procedure files include commands for selecting data tables, verifying the selection, prompting for user inputs, exporting data to ASCII files, and importing data into TransCAD. Figure 2-2 shows an example of TransCAD's interface, and how HTFS can be selected from its procedural menu. By utilizing TransCAD's

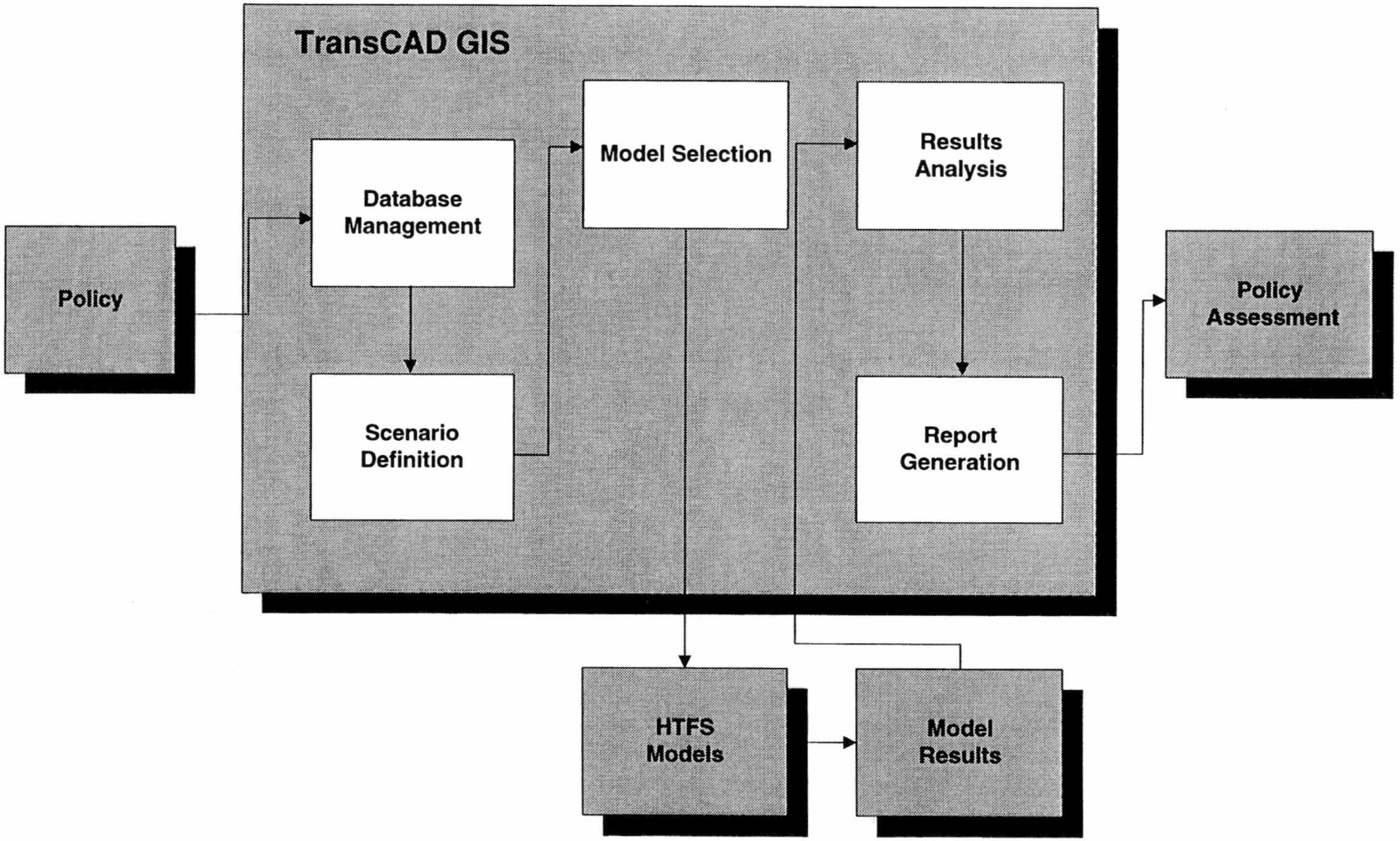


Figure 2-1: TransCAD's Relationship with HTFS
(Adapted from Ray, 1992)

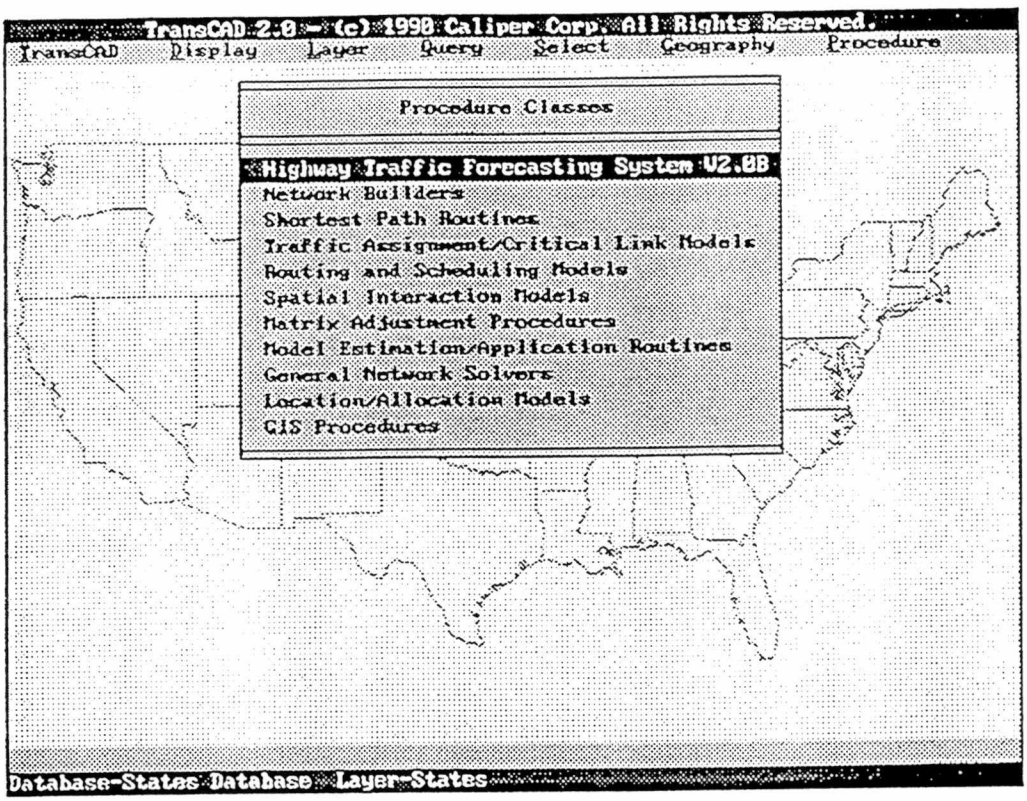


Figure 2-2: TransCAD's Interface and Procedure Selection Menu

procedure files, HTFS can guide users through the models and routines, aid the user in selecting appropriate files, verify the user's choices, build scenarios, and run the models. When the user chooses to run the models, the procedure files are used to export data from TransCAD's DBMS to HTFS's models. While the models are running, HTFS takes control away from TransCAD and displays its own screen. HTFS's screen continuously updates information about what models are running and what is being processed. Upon completion, the screen and control is returned to TransCAD.

The data managed by TransCAD can be accessed in different ways. Data can be queried and edited through TransCAD's graphical display, allowing attributes to be edited through a pop-up menu. Data can also be accessed through TransCAD's Data Editor screen. When the Data Editor is selected, a spreadsheet format screen is displayed. Data can then be managed using typical spreadsheet functions. Data are exported to HTFS's models in two different ways. First, TransCAD's procedure files are used to dump data table attributes to ASCII files which are then read by the models. Second, several data tables are extracted directly from TransCAD by using db_Vista (Raima Corporation 1990) programming libraries. The resulting files are then read as input into HTFS modules.

Once data has been extracted and processed by the models, the results can be incorporated in the GIS using TransCAD's import routines. The results can then be analyzed using its graphical display. The results of HTFS can be displayed using color, line widths, shading and hatching, and by groups and themes, depending on the type of data table being analyzed.

TransCAD also offers a number of editing tools that allow HTFS's data tables to be managed and updated as required. Transportation networks are dynamic so it is often necessary to add, delete, or realign routes. TransCAD's editing tools allow these tasks to be performed through the graphical display. Changes such as adding a route can be accomplished by using a mouse to point to where the route begins, ends, and define the shape.

The features and functionality a GIS offers has been beneficial to users in defining scenarios and running HTFS's routines. An overview of the models, major routines, and data tables of HTFS are discussed in the following section.

2.4 MODELS AND ROUTINES

In order to model freight movements on the national highway system, several different models and routines are used. Most of the modeling performed by HTFS's procedures is concentrated on the production of truck freight flows on highways, and utilizes Bureau of Economic Analysis (BEA) regions for sources of freight supply and demand (Ray and Southworth 1992). FNET uses four mathematical models: shortest path (PATH), spatial interaction (SPIN), mode split (MODESPLT), and truck competition (TRCKSPLT); and five main sub-routines: vehicle miles of travel (VMT), operating gross weight (OPWT), link tons and link limits (LINKLOAD), reconciliation module (POST/IPF), and a network building routine (NETBUILD), in addition to many supporting routines (Ray 1992). Figure 2-3 shows the relationship between the main routines and input/output files. An overview of these models and routines is given in the

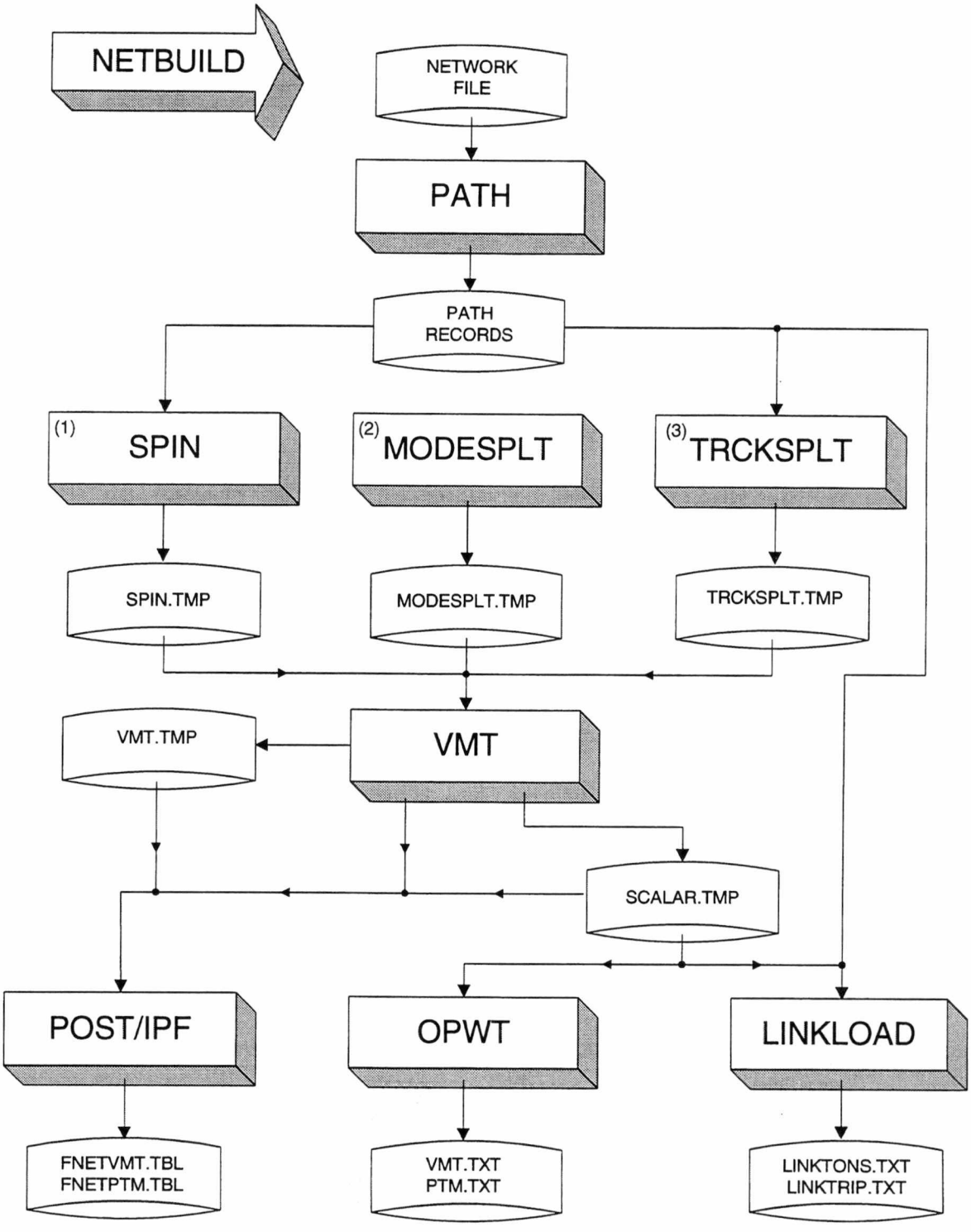


Figure 2-3
The Relationship Between HTFS's Main Routines and Input/Output Files
(Adapted from Ray, 1992)

following sections. For a more in-depth discussion of the models, refer to the Freight Network Policy Model Technical Description (Ray 1992).

2.4.1 PATH

Shortest path routines are used to determine the shortest path on a network between an origin and destination based on a function of cost. HTFS's PATH routine is a bi-objective minimum disutility path routine based on a heuristic developed by Ray (1990b) which is based on Dijkstra's (1959) shortest path algorithm. The bi-objective path algorithm is used to determine the shortest path between BEA regions while maximizing the minimum weight carried on the path and is complicated to solve. The problem is known as a "bottle neck" problem because one link along a path may affect the total quantity of cargo which can be shipped along it. For example, if a chain of four links in a path can carry 100, 80, 60, and 100 tons respectively, then a truck carrying cargo along the path is restricted to carrying 60 tons, the minimum weight allowed on the path.

The PATH algorithm calculates the generalized cost of operating a truck of class v over channel $[i,j]$ as:

$$C_{ijv} = \underset{p \in P_{ij}}{Min} \left\{ \left[(\tau) CPTM(g(p,v)v) + (1-\tau) CPM(v) \right] d(p) \right\}$$

In this equation, the function $g(p,v)$ returns the average weight of cargo carried along path p for a truck of class v, $0 \leq \tau \leq 1$. $CPTM()$ is the cost-per-ton-mile of a vehicle of class v

at gross operating weight $g(p,v)$, a non-increasing real-valued function. The function $g(p,v)$ is evaluated over the links of path p, so it can be written as:

$$g(p,v) = \underset{l \in p_l}{Min}\{OpWt_{lv}\}$$

where $OpWt_{lv}$ describes the average weight of freight moved along link l for a truck of class v, and p_l is the set of links in p. $CPM()$ is the operating cost-per-mile for vehicle type v, a non-negative real valued function of v. When $\tau = 1$, the cost is entirely a function of cost-per-ton-mile, and when $\tau = 0$, the cost is entirely a function of the operating cost-per-mile. For all other values of τ, the cost is based on a weighted combination of operating cost and gross operating weight. The equation is then multiplied by $d(p)$, which is the distance of path p (Ray 1992).

The path algorithm can be controlled in one of two ways. First, the parameter τ can be adjusted as described above. Second, the manner in which the operating weight of a truck on a link is calculated can be altered. The operating weight limits can be calculated by using a bridge or axle limits formula, or by taking the minimum of either the link operating limit or truck class maximum operating limit. The order of precedence is given to the most detailed level of data; thus greatest precedence is given to a path containing a bridge, next to axle limits, and if no data exist on these, then the minimum of the link operating limit or truck class maximum operating limit is used (Ray 1992). The input to the PATH routine is a forward-star network created in the NETBUILD routine, and is discussed in Section 2.4.5 of this chapter.

2.4.2 SPIN

Spatial interaction, or trip distribution, models are used to link trip ends (origins and destinations). The model predicts how many trips originating in Zone *i* will terminate in Zones 1, 2,...,*j*, and how many of the trips ending in Zone *j* originated in Zones 1, 2,...,*i* (Hanson 1986, p. 61). The spatial interaction models used by the HTFS are used to estimate the amount of cargo by commodity group moving between each BEA region. There are many different types of spatial interaction models and HTFS utilizes two, an Origin Constrained Gravity Model (OGCM), and an Approximately Constrained Gravity Model (AGCM). Which model is chosen is dependent on the commodity being shipped. In the commodity data table, the "Model Type" field is flagged to either a 1, specifying the OCGM, or 2, specifying the ACGM. The default model used by the HTFS is an OCGM (Ray 1992). The OCGM is used when outflow totals are known or can be predicted accurately for each BEA. When their is some uncertainty about the supply and demand totals for trip ends, the ACGM is used.

The Origin Constrained Gravity model uses an average distance measure to estimate the amount of spatial separation between BEA pairs. The average distance formula for some channel [i,j] is given as:

$$f(i,j) = \sum_{v=1,n_v} d(P_{ijv}) / n_v$$

where n_v is the number of truck classes for which a feasible channel [i,j] exists and P_{ijv} minimizes the cost from origin *i* to destination *j* for truck class *v* and channel [i,j], and *d* is the distance from *i* to *j* (Ray 1992). The origin constrained model is given as:

$$T_{ijk} = A_{ik}O_{ik}D_{jk}f(i,j)^{-\beta k}$$

where T_{ijk} is tons of commodity *k* moving from BEA region *i* to BEA region *j*. The parameter O_{ik} is the amount of commodity *k* originating from BEA region *i* and the parameter D_{jk} is the amount of commodity *k* arriving at BEA region *j*, and

$$A_{ik} = \left[\sum D_{jk} f(i,j)^{-\beta k}\right]^{-1} \quad \forall\ i \text{ and } k$$

Aik is termed a balancing factor and ensures that the summation of trips from *i* to *j*, for all *j*, balance with the supply leaving O_i, for all *i*. The beta parameter (β_k) was calibrated by using an iterative proportional fitting algorithm based on Hyman's (1969) algorithm (Ray 1992).

For some commodities, the Approximately Constrained Gravity Model is used. Also known as a Quasi-Constrained Gravity Model, this model is based on a doubly constrained gravity model and is used if there is some uncertainty about the supplies and demands at trip ends. The user is able to parameterize supply and demand constraints allowing the model to calculate the portion of flow which is not constrained (Ray 1995). If commodities are not able to be calibrated using either the OCGM or ACGM, an aggregated form of the original Reebie (1991) TRANSSEARCH database is supplied. This is an origin-destination-commodity data table containing a BEA to BEA matrix aggregated to give O_i and D_js for each commodity.

2.4.3 MODESPLT

Mode split models are used to determine the probability of a commodity taking a certain mode of transport. In order to determine the proportion of cargo that is being taken by rail or truck, a rail-truck competition modal split model is used. The modal split model utilized by the HTFS is based on a logistical model after Southworth (1978). This model determines the rail-truck split at 100 mile intervals for each commodity group. The model is given in the form of a binary logit model as:

$$P_{tk}^{road} = 1/1+e^{[-\gamma_k(\sigma t+\lambda_k)]} \quad \forall \text{ t and k}$$

In this equation, γ^t is the difference in modal cost ($C_t^{rail} - C_t^{road}$) for trips with distance interval *t*. C_t^{road} is the generalized ton-mile cost for road with trips within distance intervals of *t*. C_t^{rail} is the generalized ton-mile cost for rail with trips within distance intervals of *t*. The σ_k parameters are modal bias terms which model unquantifiable components such as measures of convenience, delivery guarantees, and safety, for each commodity *k*. The λ_k parameters depict the sensitivity of modal selection to the cost difference for each commodity *k* (Ray 1992).

2.4.4 TRCKSPLT

If a shipment is made by road, a second type of mode split model is used to determine which class of truck the shipment will take. A truck competition model is used to calculate the probability of truck class *v* carrying cargo from origin *i* to destination *j*,

and is based on determining changes in relative truck operating costs. A utility function is first calculated for each truck class for a given origin-destination pair. The utility value is then passed to a logit model which estimates the probability of use. The calculation of utility is given as:

$$C_{ijv} = \underset{P \in P_{ij}}{Min} \left\{ [(t)CPTM(g(p,v)v)+(1-t)(CPM(v)]d(p)-d(p)\sigma_v \right\}$$

The utility is calculated for each truck class v where σ_v is the dollar per mile inertia factor representing the estimated marginal change in cost, after which a shipper would switch to a different truck class. A logit model is then used to calculate the probability of selecting a truck of class v. The logit calculation is given as:

$$P_{ijv} = e^{[-\lambda C^*_{ijv}]} / \sum_{v} e^{[-\lambda C^*_{ijv}]}$$

P_{ijv} is the probability that a truck of class v will be chosen to ship cargo between origin i and destination j (Ray 1992).

2.4.5 Major HTFS Subroutines

VMT and PTM

Vehicle miles of travel (VMT) and payload ton miles (PTM) are calculated by determining the average weight of cargo shipped by each vehicle class for each channel. After the minimum cost path is calculated with the PATH algorithm, commodity flows are initially allocated to the minimum cost path from origin *i* to destination *j*. The amount in tons of commodity k traveling by truck class *v* is then calculated. After the number of tons for each movement has been calculated, the VMT and PTM is determined by deriving the number of trips necessary by each truck class to ship its share of the commodity. To do this, the operating weight distribution is used to determine the average cargo load for a truck of class *v* (see Ray 1992, Section 3.3.4). The VMT is calculated by dividing the tons of commodity k being shipped by truck class *v* from origin *i* to destination *j* ($TONS_{ijkv}$) by the average cargo load for each truck class *v* and multiplying through by the distance of the path. PTM is then calculated by multiplying the VMT by $TONS_{ijkv}$. The VMT and PTM are then aggregated to the state level.

POST/IPF

The VMT and PTM totals are disaggregated by state and truck type by road using an Iterative Proportional Fitting (IPF) routine developed by Peterson (1984). The IPF routine is used to create consistency between VMT and PTM totals by ensuring that cross-classification of these variables will sum to the correct marginal totals (Ray and

Southworth 1991; Ray 1992). For example, Figure 2-4 shows how the forecasted VMT, a two dimensional matrix of VMT by state and vehicle type, is reconciled with the basecase VMT table, a three dimensional matrix of VMT by state, vehicle type, and road class. The marginal VMT by road class in this figure is a one dimensional matrix of the observed VMT by road class. The marginal VMT by road class is a constraint which the IPF uses to assure the original distribution of VMT over road class is maintained in the forecasted VMT table.

OPWT

Operating gross weight intervals by 5000 lb. intervals is calculated by disaggregating VMT and PTM values for state, truck class, and road class by weight distribution probabilities. The maximum operating weight for each truck class is used to shift the truck distribution if necessary. The VMT and PTM is then calculated by multiplying VMT or PTM by the probability truck class v will have a maximum operating weight in a certain 5000 lb. weight interval.

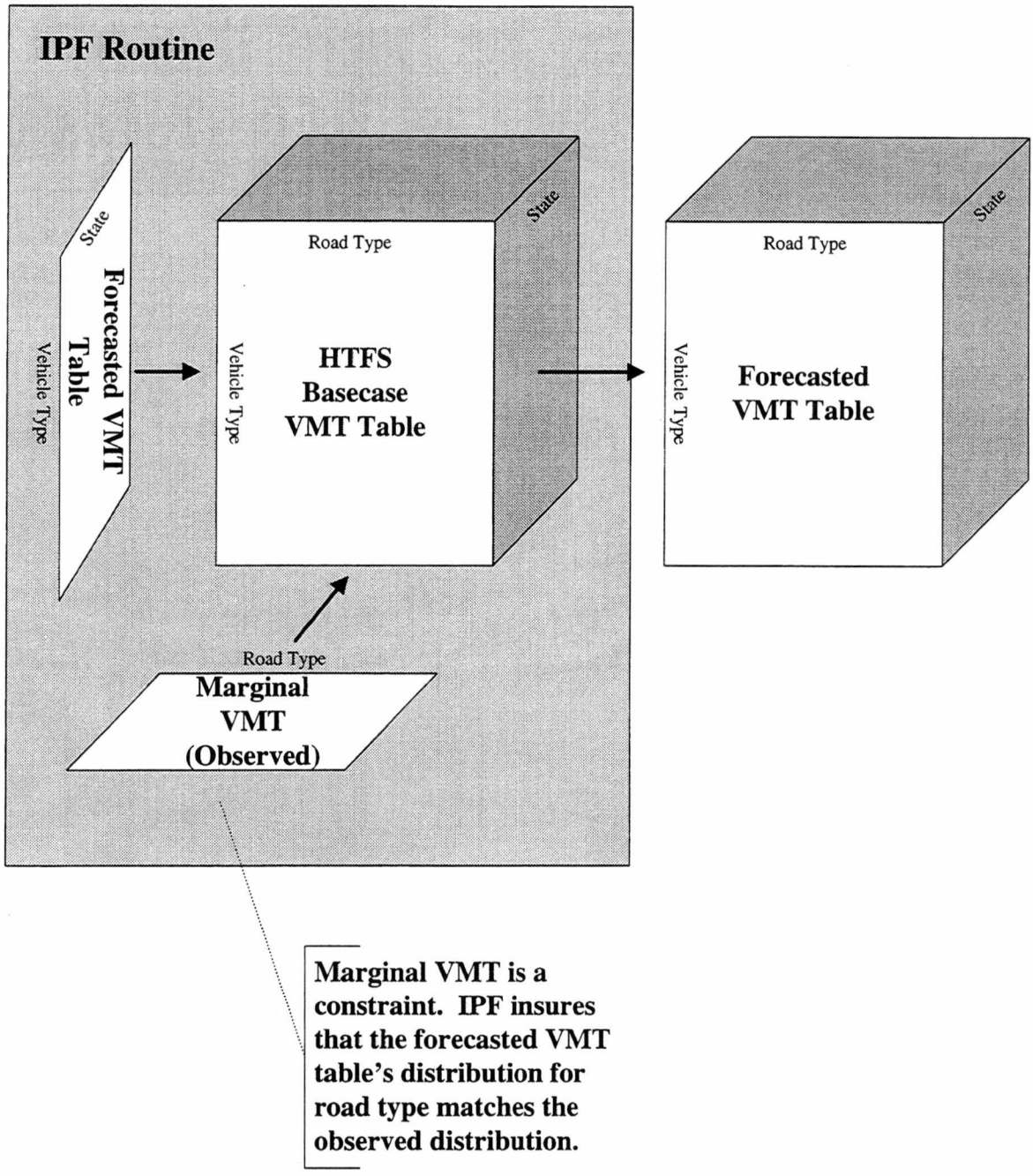


Figure 2-4: Iterative Proportional Fitting Routine
(Adapted from Ray, 1992)

LINKLOAD

The tonnage and number of trips carried by each link is determined by calculating the total number of trips and total tonnage using each link in the minimum cost path from each origin to each destination. The output of LINKLOAD is a 2-dimensional matrix of truck trips (linktons.txt), and a similar file is created for tons for link ids and truck class (linktrip.txt) (Ray 1992). These files can be imported into TransCAD or a spread sheet for further analysis.

NETBUILD

The NETBUILD routine is used to build a forward-star format network from the network stored in the GIS. A forward-star network is a compact mathematical form of the network and is used to increase the speed of processing the network by the models. The input to this routine is an ASCII network file which is exported by TransCAD. The output of the routine is a forward-star file which is read by the models (Ray 1992).

2.5 HTFS DATA TABLES

Currently, the data tables used by HTFS exist as either ASCII or TransCAD data tables. There are ten input data tables used by HTFS, five are managed by TransCAD: Network, States, BEA, Trucks, and Commodity Parameter; and five ASCII files: Commodity Supply, Commodity Demand, Operating Weight Distribution, Rail Cost, and Economic Activity data tables. Each are discussed in more detail below.

National Highway Planning Network

The current network data table is a 1:2,000,000 abstraction of Oak Ridge National Laboratory's National Highway Planning Network (NHPN) Version 1.0. This network consists of approximately 6400 links, and 4200 nodes. In TransCAD, this data table consists of a Node layer and a Link layer. The Node layer consists of four data fields describing the nodes on the network. The Link layer consists of 43 data fields, 13 are required by HTFS, 5 are required by TransCAD, and 25 fields can be defined by the user (Ray 1992). The extra 25 fields are built in the data table because of TransCAD's inflexible data model. After a data table has been defined in TransCAD, if users wish to add a field, they must export the data table, redefine it, and then rebuild it. By allocating extra attribute fields beforehand, HTFS avoids this situation. The link data table contains characteristics of the link such as length, speed limit, Anode and Bnode, and operating weight limits. The node data table contains the location and name of the node. A description of the node and link data tables items is found in Appendix A-1 and A-2, respectively.

States Data Table

The States data table is a 1:2,000,000 representation of the 48 coterminous States and Washington D.C. In TransCAD, this data table consists of 37 data attribute fields. Two data fields are required by TransCAD, nine are required by HTFS, and the rest may be defined by the user (Ray 1992b). The data table contains information on allowable weights for different truck configurations and sub-network permissions (i.e., what truck

classes are allowed to operate in the state). A description of the data table items is given in Appendix A-3.

Bureau of Economic Analysis Data Table

The BEA data table is a 1:2,000,000 representation of the 181 conterminous Bureau of Economic Analysis regions in the nation. In TransCAD, this data table contains 47 fields. Two data fields a required by HTFS, and four fields contain socio-economic data from the 1986 County Census Patterns Database; the remaining fields can be defined by the user. This data table is not required by HTFS but facilitates defining and analyzing some of the data required by the models (Ray 1992b). A description of the data table items is given in Appendix A-2.

Trucks Data Table

The Trucks data table is an aspatial TransCAD data table. The data table contains specifications, costs, and parameters for models using truck characteristics. The data table contains 18 fields. 9 fields are required by HTFS (e.g., operating costs and maximum weight limits) and the remainder can be defined by the user (Ray 1992b). A description of the data table items is given in Appendix A-4.

Commodity Parameter Data Table

The Commodity Parameter data table is an aspatial TransCAD data table. The data table contains the parameters for the spatial interaction and modal split models.

These parameters have already been calibrated, and should only be changed to reflect observed distributions in the data (Ray 1992b). A description of the data table items is given in Appendix A-5.

Operating Weight Distribution Data Table

The Operating Weight Distribution data table is an aspatial ASCII data table containing weight distribution data for each truck configuration in the Trucks data table. The data includes an empirically-derived probability distribution for each truck type. This defines the probability of a truck in a given class operating in a given 5000 lb. weight interval. Other data fields include probability distributions for empty and loaded weight over each axle group, an estimated average unloaded weight for the truck class, and the maximum gross weight rule which the distributions were derived under, normally 80,000 lbs. (Ray 1992b).

Commodity Supply Data Table

The Commodity Supply data table is an aspatial ASCII data table representing total supply for every BEA region for each commodity moved by truck or rail modes. Each tuple in the data table contains a BEA, COMMODITY, and SUPPLY field (Ray 1992b).

Commodity Demand Data Table

The Commodity Demand data table is an aspatial ASCII data table representing the total demand for every BEA region for each commodity moved by truck or rail modes. Each tuple in the data table contains a BEA, COMMODITY, and DEMAND field (Ray 1992b).

Rail Cost Data Table

The Rail Cost data table is an aspatial ASCII data table containing rail cost per ton-mile in dollars divided into 100-mile distance intervals (Ray 1992b).

Economic Activity Data Table

The Economic Activity data table is an aspatial comma-delimited ASCII data table containing a list of economic growth parameters for each commodity for each BEA region (Ray 1992b).

Summary

This chapter has described the history of HTFS, its current use within TransCAD, and an overview of its major modeling routines and data tables. If the user wishes a more in-depth technical description of the models and routines, he should refer to the Freight Network Policy Model Technical Description (Ray 1992). The next chapter will give an overview of Arc/Info, and two software systems running in Arc/Info, which can be used to help define and analyze HTFS's scenarios.

CHAPTER 3

INTRODUCTION TO ARC/INFO, NETSHELL AND TED

Arc/Info, developed by Environmental Systems Research Institute, Inc. (ESRI), is one of the most widely used GISs in the world. A key feature in making Arc/Info so popular is its flexibility. Arc/Info is a modular system which enables users to access separate modules for many diverse uses. Its basic modules are Arc, Arcplot, Arcedit, and Info. Arc is the foundation of Arc/Info and is used to run many of its spatial analysis routines; Arcplot is used to display, query, analyze, and create maps from spatial data tables; and Arcedit is used to create and edit spatial data tables. Info is a relational database management system (DBMS) used to manage both spatial and non-spatial data, although many different DBMSs can be connected to and utilized by Arc/Info.

In addition to these modules, ESRI offers modules designed for surface modeling, cell-based modeling, and network analysis, allowing Arc/Info to be used to analyze a great variety of issues. Arc/Info also has a powerful Arc Macro Language (AML) which can be used to create custom Graphical User Interfaces (GUIs), batch process commands, and perform many other functions.

AML was used to develop NETSHELL and the Topological Editing System (TED). Both systems were developed by the GIS Group at the University of Tennessee's Transportation Center. These systems are tailored to transportation issues, and both systems can be used to help define, manage, and analyze HTFS's scenarios. NETSHELL is an Arcplot GUI, and was developed for network design, analysis, and management. It has a comprehensive set of network management, modeling, display, and mapping

routines. TED is an Arcedit GUI, and was developed for network editing and management of network attributes. Together, NETSHELL and TED encapsulate the majority of commands required to manage, create, and modify network data tables through comprehensive GUIs; therefore, HTFS was designed to be invoked from either of these systems. Being able to execute HTFS from these systems is beneficial to the user because, unlike TransCAD which has many of the features described above accessible through a single display screen and interface, the user must exit one module and start another module to perform certain functions in Arc/Info.

The first section of this chapter will give a brief introduction to the modules and functions of Arc/Info which will be utilized by HTFS. These include Arc, Arcplot, Arcedit, Info, and AML. A description of how Arc/Info manages spatial features and topology is also presented, followed by an overview of NETSHELL and TED, their primary features, and the benefits for making HTFS accessible from them.

3.1 ARC/INFO MODULES

Arc/Info is an extremely large, powerful, and complex GIS. Therefore, only the features and functionalities pertinent to this study will be investigated. Since this section is meant as an introduction and overview of Arc/Info, many of the more detailed aspects will be introduced in the following chapter which deals with the integration of HTFS and Arc/Info.

3.1.1 The Arc Module

Arc is the basic module of Arc/Info. It is a non-graphic module and is primarily used for executing spatial analysis routines, importing and exporting data, building topology, starting other modules, and controlling the environment in which Arc/Info is operating. When HTFS is integrated with Arc/Info, the main functions of the Arc module are to build topology when a topological change is made, copy data tables, and start HTFS's GUI or to start NETSHELL or TED from which HTFS's GUI can be invoked.

3.1.2 The Arcplot Module

Arcplot is a graphical display module and is typically used to display, analyze, and create maps from spatial data, but has many additional routines which can be utilized by HTFS. Arcplot's powerful display capabilities allow features to be displayed in a number of different methods. For example, all features can be displayed in a wide range of colors; line features can be represented in an assortment of different thickness and styles; polygons can be represented in numerous shades and styles; and points can be represented in a variety of symbols and sizes. Annotation can also be added to features. Annotation may come directly from a feature's attributes, or can be transcribed by the user. The display capabilities of Arcplot facilitate the analysis of features in addition to being able to produce maps relatively quickly.

Arcplot has a large tool-box of routines for graphically selecting, identifying, and editing feature attributes. Complex spatial queries can be used to select features using logical expressions. For example, queries may include selecting all Interstates, selecting

all Interstates within a certain county, selecting all counties with an Interstate passing through them, or selecting all counties without an Interstate, but which are within a certain distance of an Interstate. Once a feature is selected, it can then be analyzed or its attributes can be edited using additional routines.

The capabilities of Arcplot makes it a sound platform for HTFS. HTFS can utilize its display, query, and editing functions for the majority of its data management. In addition, NETSHELL has automated and expanded many of these routines, thus allowing these functions to be executed through a menu interface.

3.1.3 The Arcedit Module

Arcedit is a graphical display module primarily used for editing the physical structure of spatial data tables and their attributes. Arcedit has a large number of topological editing tools designed to facilitate editing and creating spatial data tables. For example, features can be added, deleted, moved, or reshaped graphically via a mouse. In addition, Arcedit has a suite of DBMS commands which are used to manage the non-spatial attributes of a coverage, as well as powerful display capabilities. Although more limited in its selection and display routines then Arcplot, Arcedit is more intuitive to learn, and often provides all the capabilities required for defining and analyzing a scenario.

3.1.4 The Info/Tables Modules

Tables is an Arc/Info interface to Henco Software's (1991) INFO, the standard DBMS used by Arc/Info. Info uses a relational data model to manage both spatial and non-spatial tabular data. INFO is similar to other commercial DBMS in that it has many routines for selecting, viewing, and editing data in tabular format. The data tables managed by Tables can be accessed through any of Arc/Info's modules. Spatial data can be accessed in Arc/Info's graphical modules by using point-and-click methods, and both spatial and non-spatial data can be accessed through AML directives or by setting up relations between data tables. Using Tables, or entering the Info environment directly, will allow HTFS's data tables to be accessed in a DBMS environment. The benefit of this is that non-spatial queries or editing can be performed relatively quickly without having to start NETSHELL and draw its spatial features. In addition, the ability to access data tables in Tables or through other Arc/Info modules will allow HTFS's database to be managed in the most efficient manner.

3.1.5 AML

Arc/Info is a command line-driven GIS. Arc Macro Language (AML) provides programming capabilities and tools allowing commands or groups of commands, normally executed by typing them at the command line, to be batch processed by executing an AML file containing the commands. AML also offers the capabilities to build custom GUIs for Arc/Info applications; thus allowing commands to be executed through pull-down menus or by selecting buttons on a menu with a mouse. AML is an

interpreted language which means each command is translated to machine language and then executed before moving to the next line.

The basic building blocks used by AML are directives, variables, and functions. Directives are used to perform specific tasks such as setting terminal characteristics, writing messages to the screen, assigning variables, and sending messages to the operating system. Functions are used to perform a predefined operation and return a single value. Functions perform many different operations including verifying the existence of a file or coverage, prompting the user for input, performing mathematical calculations, and reading and writing files (ESRI AML 1993).

An important design consideration when developing a GUI with multiple menus is the method of retrieving information from the appropriate menu at the appropriate time. AML uses threads to perform this function. Essentially, threads are used to activate and deactivate menus and AML routines when necessary. For example, if two menus are being displayed at the same time and information needs to be retrieved from one of the menus, then an AML directive is used to activate (i.e., focus) the menu where information is required from, and deactivate (i.e., unfocus) the menu where information should not come from.

AML is used to create a custom GUI for HTFS, aid the user in selecting the appropriate files and data tables, and manage the data flow between the models and Arc/Info. For a complete discussion on AML the reader should refer to ESRI (1993). The following section will give a brief overview of how Arc/Info stores spatial features and its topological data model.

3.2 SPATIAL FEATURES AND TOPOLOGY

A coverage is the method used by Arc/Info to store point, line, and polygon geographic data sets. A coverage contains both spatial and attribute data for geographic features, and Arc/Info's coverages are georelational (i.e., the descriptive data are associated with spatial data through internal linkages).

Attribute data are stored separately from the spatial data in feature attribute tables (FATs). FATs are special tables managed by Arc/Info, and are a part of the coverage. Linkages between spatial features and their attribute data are maintained through unique internal identifiers (IDs). Each spatial feature has a unique ID which matches a unique ID in the FAT allowing the spatial feature to be linked to its descriptive data. Figure 3-1 shows a typical linkage between a spatial feature and attribute data. A number of different FATs can be created for a single coverage, and each FAT is associated with a particular feature class. FATs are named after the type of feature they represent. Arc attribute tables are named AATs, node attribute tables are named NATs, and point and polygon attribute tables are named PATs. Additional items can be added to a FAT to further describe the feature or to be used as keys to relate to other tables maintained by the user (ESRI AML 1993). The ability to relate to other tables allows data to be separated, thus helping to minimize data duplication. Figure 3-2 shows how feature attribute tables can be related to other tables.

Arc/Info uses topology to explicitly define spatial relationships between spatial features. By creating and maintaining topological relationships, data can be stored more

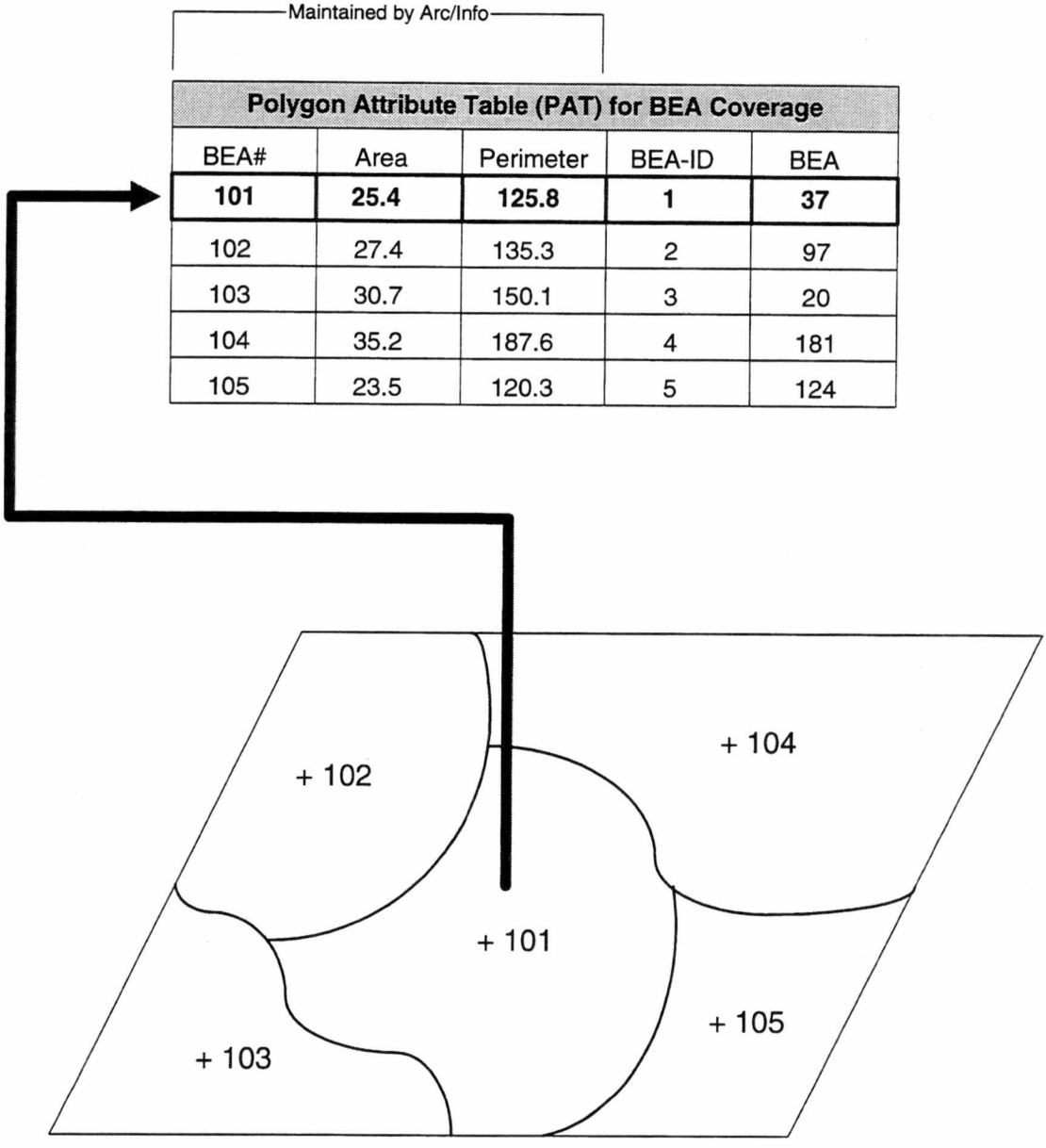


Figure 3-1
Arc/Info's Georelational Linkage for a BEA Coverage

Polygon Attribute Table (PAT) for BEA Coverage				
BEA#	Area	Perimeter	BEA-ID	BEA
101	25.4	125.4	1	37
102	27.4	135.3	2	97
103	30.7	150.1	3	20
104	35.2	187.6	4	181
105	23.5	120.3	5	124

BEA Data Table			
BEA	Population	% Employed	# Businesses
37	37,000	84	4,300
97	10,100	91	980
20	12,000	87	1,000
181	17,500	89	1,350
124	11,120	90	1,157

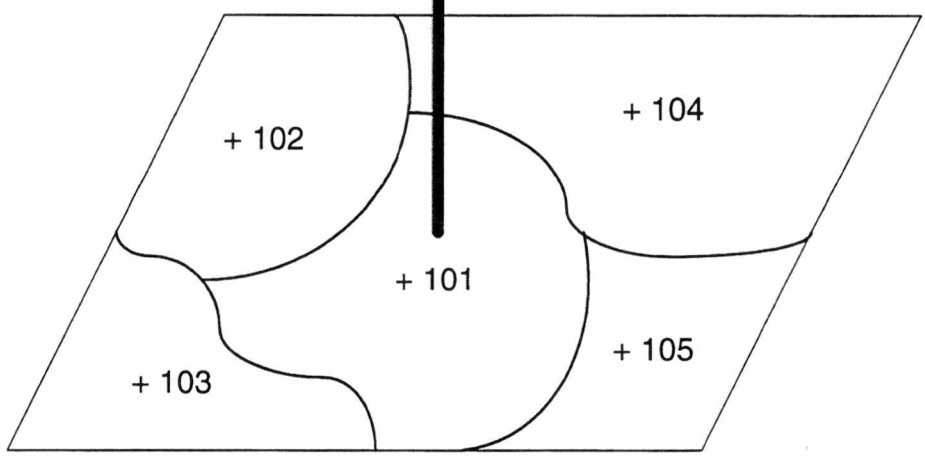


Figure 3-2
Arc/Info Relation Between a FAT and a Data Table

efficiently and the GIS's analytical routines are facilitated. Connectivity, area definition, and contiguity are the three main topological concepts used by Arc/Info (ESRI AML 1993). Connectivity assures that arcs connect at nodes. Each arc has a *from-node* and *to-node* item in its FAT which contains the unique ID of the node the arc begins and ends at, respectively. This gives each arc a direction, and allows easy identification of which arcs are connected to each other. Area definition maintains that arcs connecting to enclose an area define a polygon. By defining polygons as lists of arcs instead of (x,y) coordinates, an arc shared by two adjacent polygons need only be stored once, thereby reducing data storage. Two features sharing a common border define adjacency. Contiguity is the topological concept used to determine the adjacency of spatial features. Each arc has a *left-poly* and *right-poly* item in its FAT which contains the unique ID of the polygon to its left and right, respectively. Thus, allowing adjacent polygons to be identified (ESRI AML 1993). These items are null in the AAT if the coverage has not been built for polygons. Figure 3-3 shows the topological relationship between arcs and nodes.

3.3 NETSHELL AND TED

NETSHELL and TED are of GUI's and sets of software tools which can be used to perform many different operations, including the following: displaying spatial data thematically; querying spatial data interactively; creating, modifying, and editing spatial data tables. Additionally, NETSHELL has routines for solving various network optimization and transportation models, and displaying results of model outputs and queries. TED has a comprehensive set of network editing and management tools.

Arc-Node Topology		
Arc ID	From-Node	To-Node
1	**A**	**B**
2	B	E
3	C	D
4	D	E
5	E	G
6	D	F
7	F	G

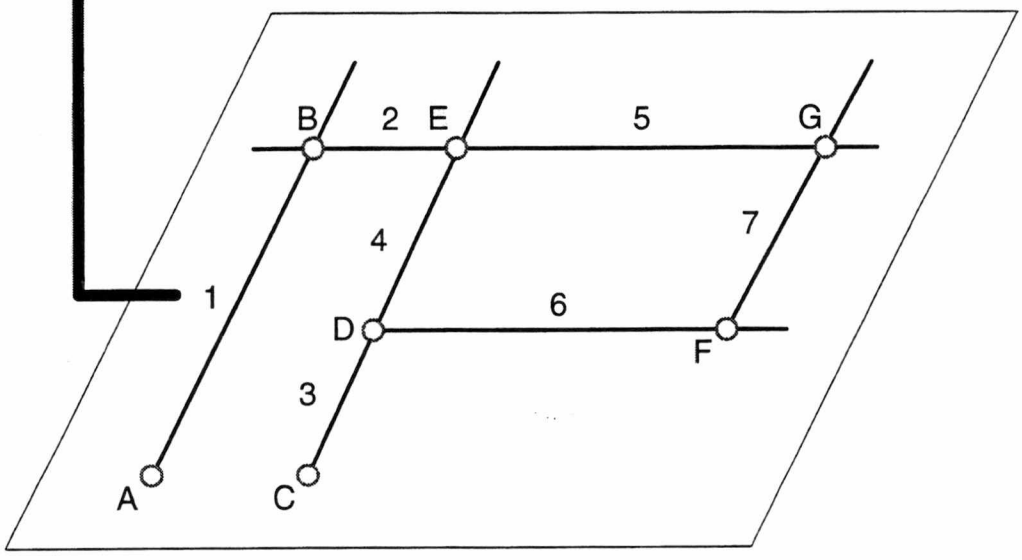


Figure 3-3
Arc and Node Topology

Both NETSHELL's and TED's GUIs consist of menus, text windows, dialog windows, an information bar, and a display window. Figures 3-4 and 3-5 show NETSHELL's and TED's GUI, respectively. Text windows are used to display information such as the results of spatial queries. Dialog windows are interacted with to make selections, enter data, and perform a number of different interactive functions. The information bar is used to display information, such as the number of features selected, and the Display Window is used for drawing coverages and selecting features graphically (HNDES 1994, TED 1995). From this point, NETSHELL and TED have different orientations, and both systems are discussed separately in the following two sections.

3.3.1 NETSHELL

NETSHELL Network Management Utility Version 1.0 was developed by Dr. Julian J. Ray and Mr. Dean M. Miller at the University of Tennessee Transportation Center (UTTC). It runs as a shell program in Arcplot and is primarily written in AML, but has a number of modeling routines written in C. NETSHELL is a superset of the Highway Network Design and Evaluation System (HNDES) Version 1.0b. HNDES was developed jointly between the UTTC and Oak Ridge National Laboratory's Center for Transportation Analysis for the FHWA's Office of Policy Development. HNDES is a software system developed as a strategic planning tool for aiding in the analysis of alternative national and inter-regional highway systems (HNDES 1994). NETSHELL has all the features and functionalities of HNDES but has had some additional modeling routines integrated into it. As of this time, no documentation has been developed for

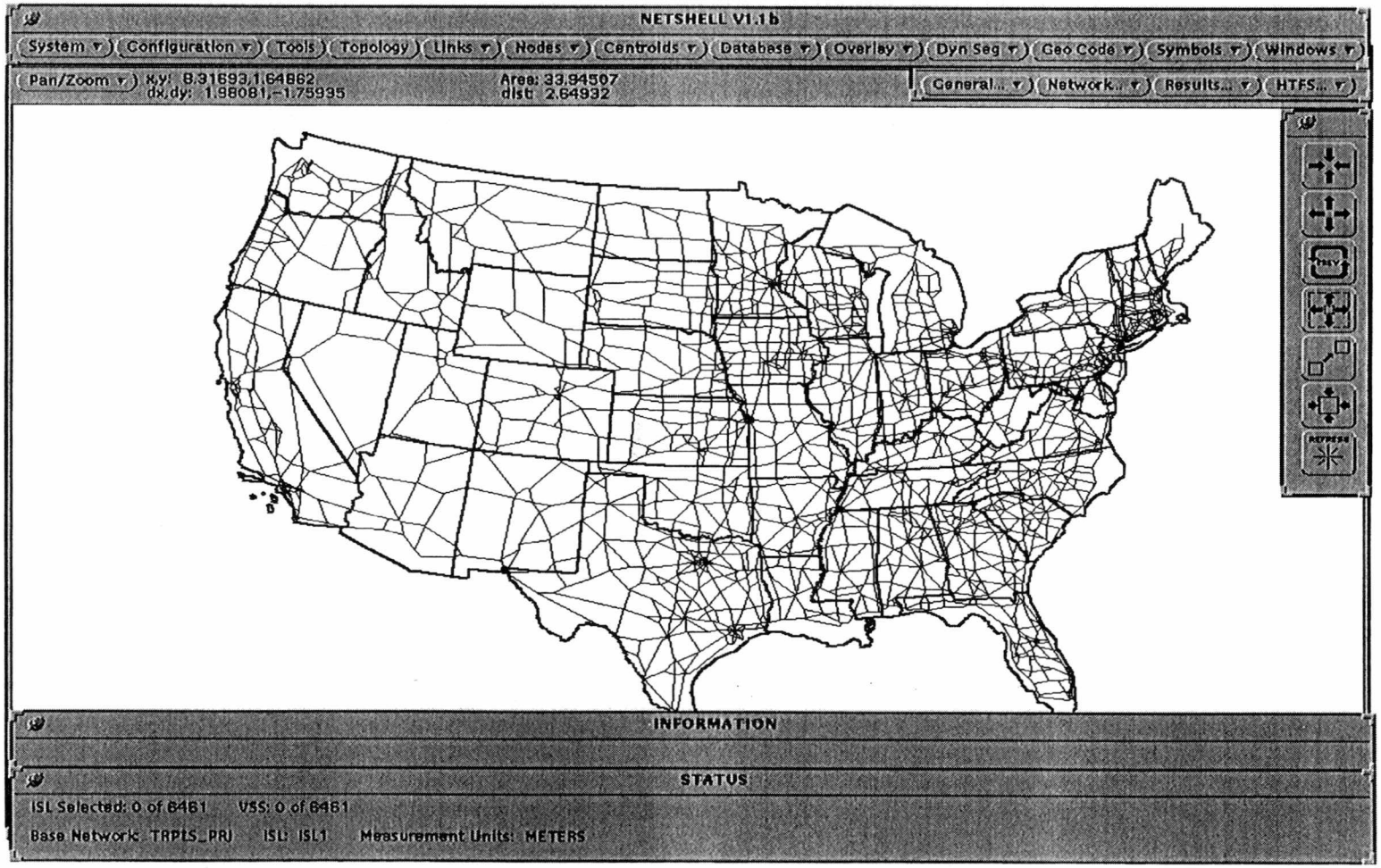


Figure 3-4: NETSHELL'S GUI

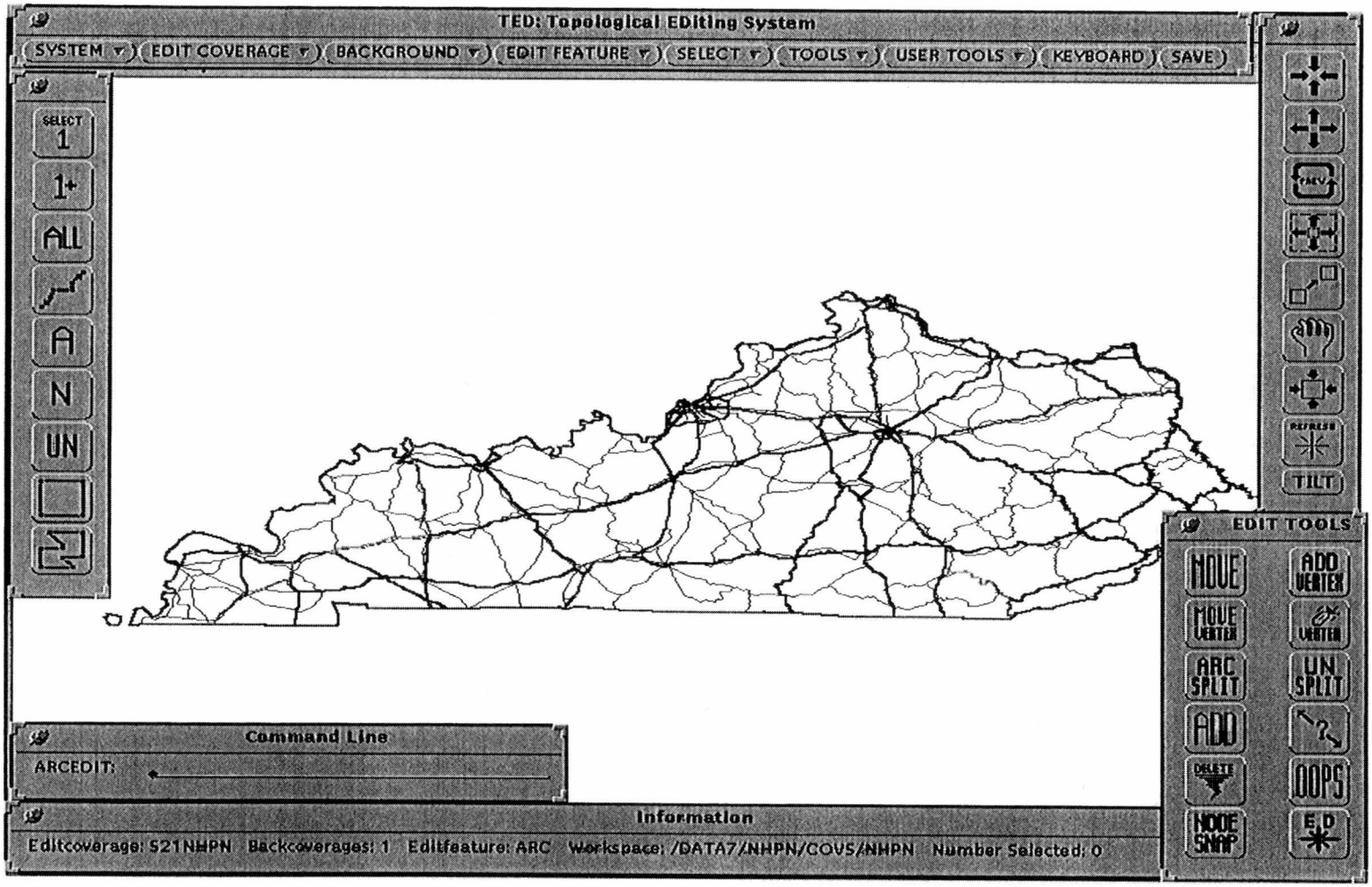


Figure 3-5: TED's GUI

NETSHELL, so the HNDES user manual is used as the source of much of the information contained in this section.

Some of the primary tools which will be utilized by HTFS are NETSHELL's drawing tools, selection tools, editing tools, and analysis routines. The drawing tools will manage the graphical environment of HTFS's data tables including the colors, styles, and symbols selected. The drawing tools also include a number of tools for zooming and panning around the data tables, and refreshing the Display Window. NETSHELL has a number of selection tools that allow features to be selected graphically, or through an expression generator. The expression generator allows features to be selected, using Boolean logic, through a menu interface. Once features have been selected, NETSHELL's editing tools can be used to update feature attributes. NETSHELL also has tools for extracting selected features to a new coverage. These routines will be useful in selecting subsets of the network, states, and BEA data tables and creating new coverages from them. NETSHELL also has routines which allow the results of model runs to be analyzed in an efficient manner. For example, NETSHELL has a routine for loading determined flows onto the network and displaying these flows thematically.

NETSHELL will provide most of the tools required to manage HTFS's database, draw features, update attributes, and display results. In addition, NETSHELL has a large number of supplementary tools not discussed above, but which can be utilized for different types of analysis not related to HTFS. This will allow HTFS's database to be utilized for other purposes. These tools include a number of models for testing network

connectivity, determining trip distribution, and a number of shortest path models. In addition, all of Arc/Info's standard modules can be accessed from NETSHELL.

3.3.2 TED

The Topological EDiting System (TED) Version 1.0 was primarily developed by this author at the University of Tennessee Transportation Center. It runs as a shell program in Arcedit and is primarily written in AML, but has some supporting FORTRAN routines. TED is designed to provide easy access to Arcedit's many editing commands and has the ability to trace and historically log edits made to a network. TED encapsulates the majority of commands required to edit spatial features, and has a number of graphical display capabilities which can be used to thematically display spatial data. TED can be used to graphically edit both the physical structure of spatial data tables, as well as their non-locational attributes. In addition, TED's GUI is fully customizable by the user, and has the ability to run any user routines through its pull-down menus (TED 1995).

Since many of the scenarios HTFS can analyze require topological changes to the network or its attributes, TED can be used to efficiently make these changes, and then immediately run HTFS and graphically view the results. Although TED does not have the powerful network analysis routines which NETSHELL encompasses, its simple display and editing routines often make it a preferable platform from which to execute HTFS.

Summary

This chapter discussed the primary modules HTFS will utilize within Arc/Info (i.e., Arc, Arcplot, Arcedit, and Tables), Arc/Info's data structure, and an overview of AML, which will be used to create the integration of HTFS and Arc/Info. In addition, an overview of NETSHELL and TED was presented and the reasons they are beneficial to HTFS. The following Chapter will describe the issues in porting the HTFS to a Unix platform and discuss the procedures used for integrating HTFS with Arc/Info.

CHAPTER 4

INTEGRATION OF HTFS AND ARC/INFO

Geographic Information Systems cannot contain all the specialized models and routines needed for transportation analysis. Consequently, users are integrating their own models with commercial GIS. This chapter will focus on the steps and procedures developed to integrate the Highway Traffic Forecasting System with Arc/Info. The chapter begins with an overview of the procedures used for the integration of HTFS and Arc/Info, followed by a discussion of the design and development of the AML code used to integrate HTFS with Arc/Info.

4.1 PROCEDURES FOR INTEGRATION

Several steps were taken in the porting and integration of HTFS. The major steps of the integration of HTFS and Arc/Info were:

- Developing a comprehension of the original design parameters and usage of HTFS.
- Porting and translating TransCAD's data into Arc/Info's data structure.
- Compiling and modifying HTFS's models and routines for the Sun workstation.
- Testing and debugging HTFS's models and routines outside the GIS.
- Replicating the original data formats and data flow in Arc/Info.

Modifications to HTFS's source code were made only when absolutely necessary to minimize the possibility of corruption.

4.1.1 Original Design Parameters and Usage of HTFS

The first step in the integration of HTFS and Arc/Info was to develop a strong understanding of how HTFS acts within its original design parameters. A working knowledge of the system was developed through installing, testing, and running HTFS under TransCAD. HTFS does not have an installation program so the user is required to create the directory structure and copy the appropriate files to their proper directories. For a first-time user, the lack of an installation routine aids in developing a foundation in HTFS's directory structure and the organization of HTFS's files.

HTFS was designed to run under TransCAD Version 2.0, but this version of TransCAD was not available, so GIS/Plus was used for the testing. GIS/Plus was also developed by Caliper Corporation, and is identical to TransCAD, except that some of their transportation modeling routines are not included. According to GIS/Plus's documentation, the procedural language (macro language) of TransCAD and GIS/Plus are identical; therefore, because HTFS does not use any of their modeling routines, HTFS's should work identically in either software package. However, this was not the case, as undocumented changes had occurred to GIS/Plus' procedure's syntax. The main change that affected HTFS's procedures was that all output file names in the procedures needed to be enclosed in double quotes. This change was made to all of HTFS's procedure files, and HTFS was then able to run under GIS/Plus[1].

After HTFS was installed and running under GIS/Plus, a number of scenarios were created and run, mostly based on HTFS's tutorial, to develop a familiarization with

[1]An identical undocumented change took place in TransCAD Version 2.01.

its interface and usage. HTFS's different options were examined by running existing scenarios, creating new scenarios, and extracting the network. The intent of this process was to become familiar with its original usage, features, and functionalities so that they could be recreated as closely as possible in Arc/Info.

4.1.2 Porting and Translating Data

HTFS' data tables needed to be ported to the Unix workstation and imported into Arc/Info. TransCAD has several export options. For this thesis, TransCAD's TCBuild module was used to export the data tables into comma delimited ASCII format files. TCBuild's export function unloads the data into one or more different ASCII files. For a line data table, line coordinates and attributes are exported to a file with a .DL extension, and node attributes and coordinates are exported to a file with a .D0 extension.

All files, even polygons, were exported to ASCII as lines. This was done because when a polygon coverage is exported from TCBuild as polygons, TransCAD writes each area, which may be represented by multiple polygons, as a single line. Two main problems can occur in Arc/Info by representing polygons as a single line. First, when a polygon coverage, where each polygon is represented as a single line, is imported into Arc/Info, two contiguous polygons will have their shared border duplicated. Arc/Info, however, represents the boundary of contiguous polygons as a single non-duplicated line. Duplication can cause problems when building topology in Arc/Info because Arc/Info's 'BUILD' command ensures planar enforcement, which means that polygons cannot overlap. A related problem is that the resolution of the data may have been altered

somewhat when imported into Arc/Info, thus causing vectors of the duplicated lines to intersect. This change of resolution can occur because of rounding error when the coordinate values are imported. Although this situation can be remedied using Arc/Info's 'CLEAN' command, which puts a node at the intersection of two arc vectors and merges arcs within a given tolerance, it may be necessary to edit numerous "sliver" polygons. Figure 4-1 shows an example of a network coverage before and after its is *cleaned.*

The second problem encountered when exporting polygons from TransCAD for import into Arc/Info is some areas may be represented by more than one polygon. For example, in the States data table, California is represented as a number of polygons. One polygon represents California's mainland, while a number of other polygons represent its off-shore islands. TransCAD represents all of California's polygons with a single line. When TransCAD exports areas with multiple polygons, all the separate polygons of an area are bundled into a single list of coordinates, which define the shape of its boundary. If this coordinate list is imported into Arc/Info, lines will be included connecting all of the area's polygons. Figure 4-2 shows an example of California, its external polygons, and the resulting lines connecting all polygons. These polygons can be parsed out in a translation utility or residual editing can be performed to correct the data table. However, both of these tasks are time consuming and unnecessary if the polygons are exported as lines.

These problems were avoided by exporting the TransCAD data table as lines. Exporting polygons from TCBuild with the line option ensures that areas with multiple polygons will not be exported as single lines, thus enabling the use of relatively

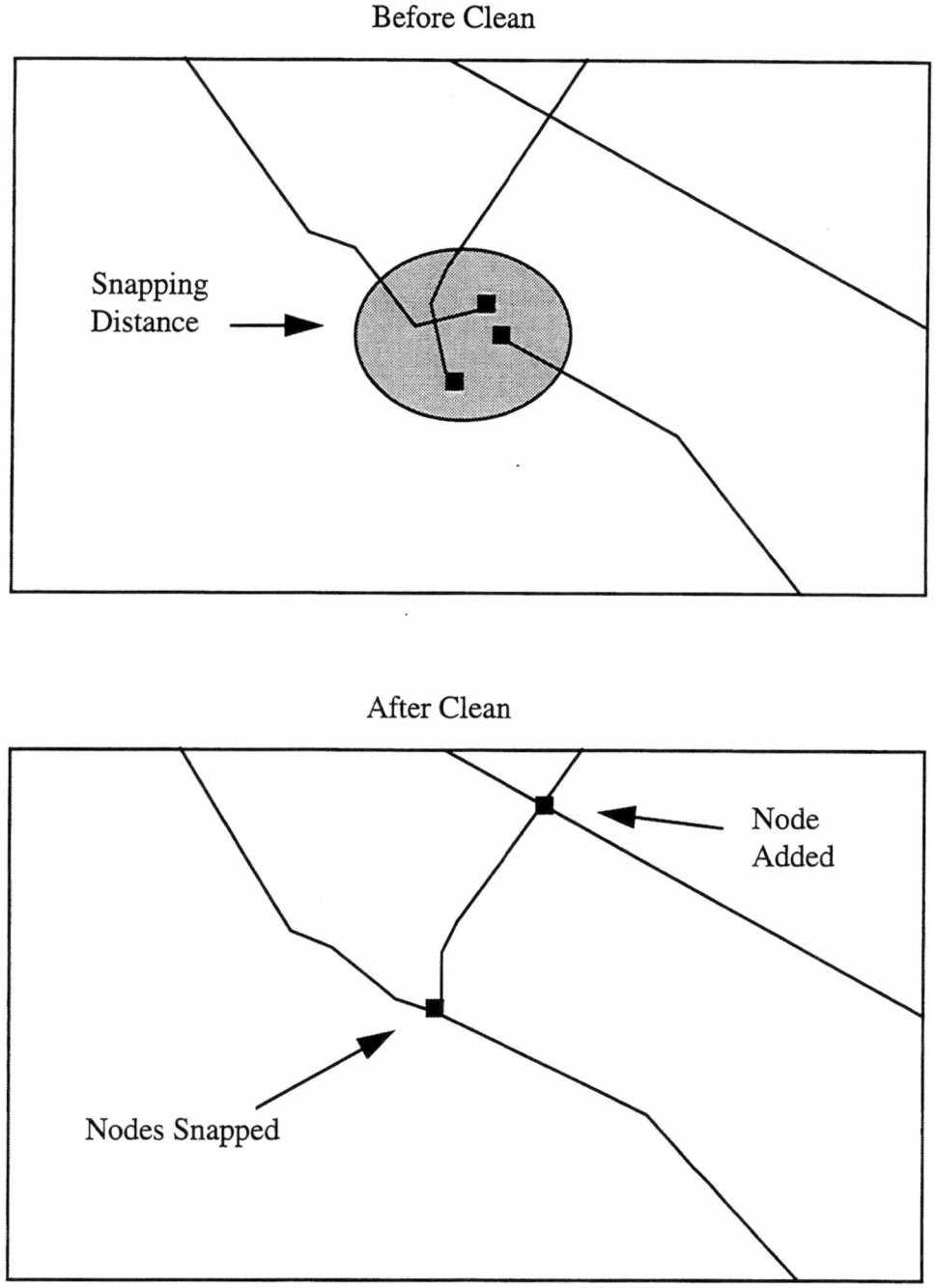


Figure 4-1: Example of Arc/Info's CLEAN Command

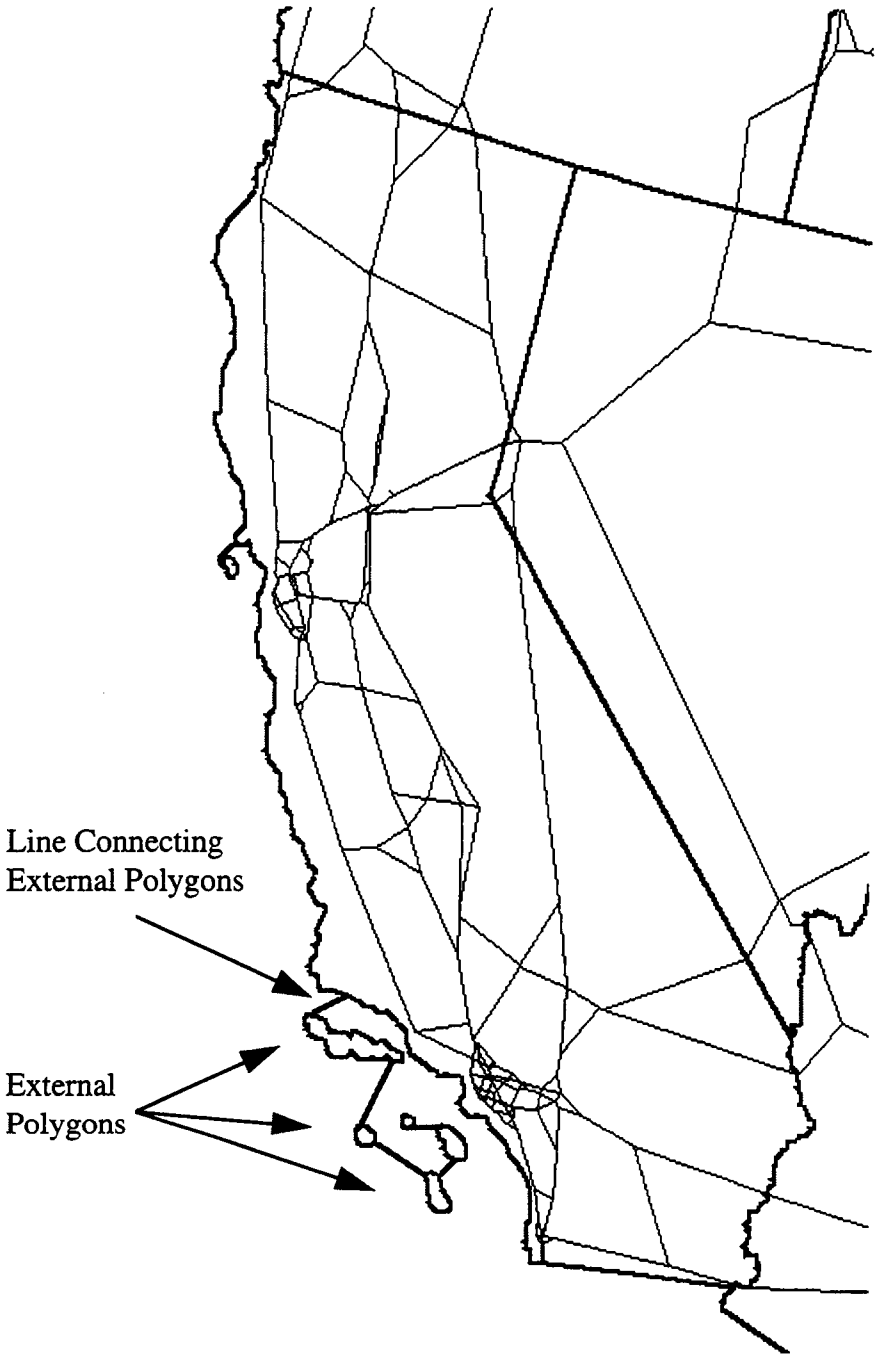


Figure 4-2: Example of External Polygons in California

simple translation utilities and requiring little or no residual editing once imported into Arc/Info.

The format which TransCAD exports the data into is not readable by Arc/Info; therefore, C routines were developed (Appendix B) to reformat the ASCII data into Arc/Info generate format files and comma delimited attribute files. Generate format files are specifically defined format files which Arc/Info reads to import locational (coordinate) data. The translation routines were used to translate the National Highway Planning Network Version 1.0 (NHPN V. 1.0) data table, the BEA data table, and the States data table. Figure 4-3 shows an example of a TransCAD export file, and the resulting Arc/Info generate format file and attribute file after the translation routines are run. Centroids from the polygon coverage were extracted from TransCAD using an extraction routine developed by Caliper Corporation, and imported into Arc/Info as points, which were then converted to labels for the polygons attributes.

After the coordinate data were converted to generate format files, the spatial data were imported into Arc/Info using the 'GENERATE' command, and topology was built using the 'BUILD' command. AML routines were developed to import the attribute data into temporary Info tables. The next step was to associate the attribute data with their locational features. To link the attributes with the locational features, Arc/Info's 'JOINITEM' command was used. This command creates a user-defined relation between an ID in the spatial feature's attribute table and an ID in the temporary Info file; after the relation is established, the temporary Info file is appended to the feature attribute table. For the NHPN, the nodes were imported as points, and then Arc/Info's

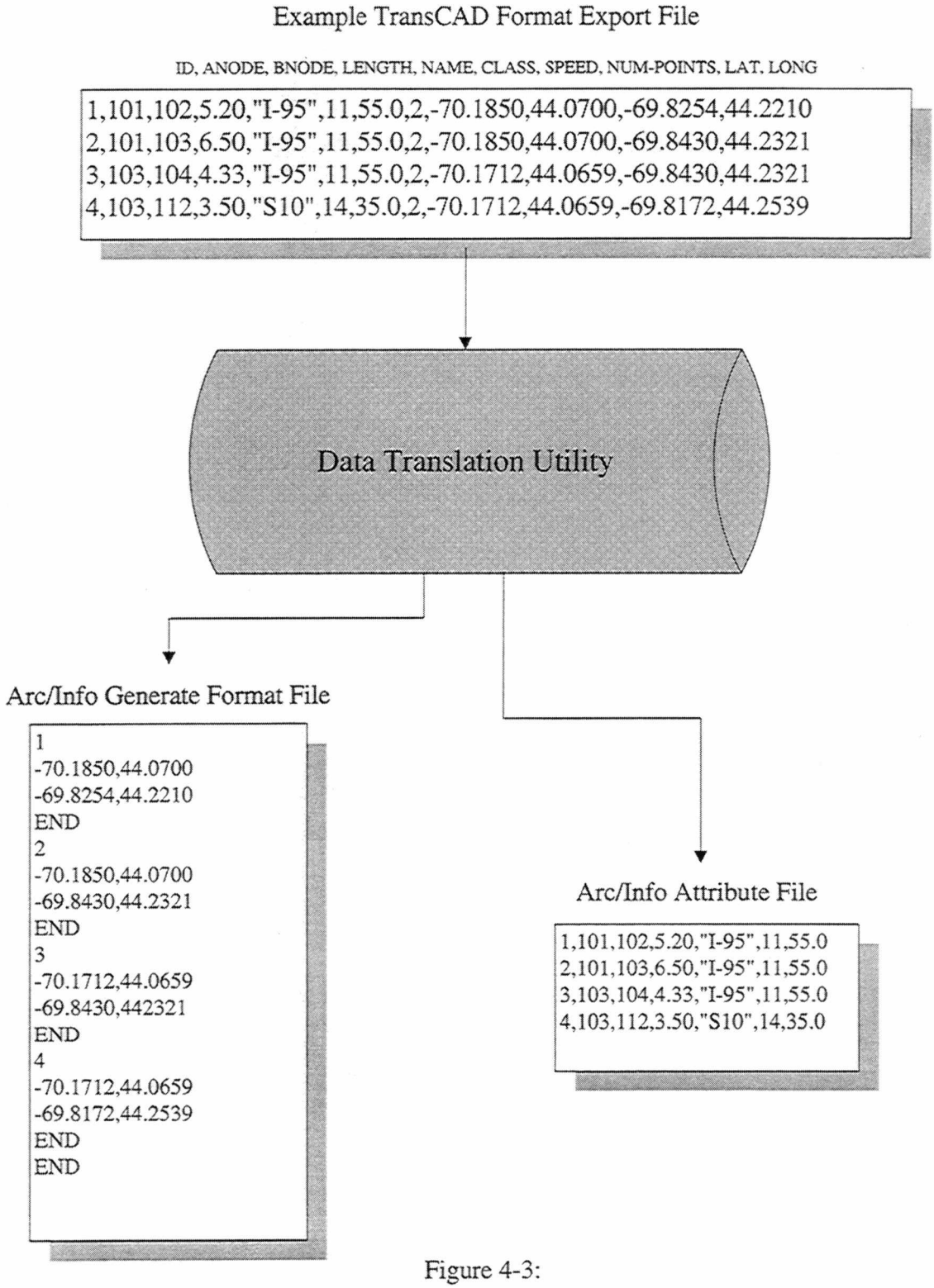


Figure 4-3:
Example of TransCAD Data Translation

'POINTNODE' command was used to transfer the point's (literally node's) attributes to the nodes. For the polygon coverages, centroids were generated at the same time as their associated lines, they were built as polygons, and then their attributes, which were imported into an Info file, were joined with their attribute table with the 'JOINITEM' command. Two non-locational data tables, the truck and commodity data tables, were imported into Info from the comma delimited export files TransCAD wrote; however, because they have no links to spatial features, they were left as Info files.

4.1.3 Compiling HTFS's Models and Routines

HTFS's models and routines were engineered with the intent of having them integrated with a GIS running on a Unix workstation at a later date. For this reason, the code had been designed and tested on a Unix platform. There were two main design considerations that ensured the proper compilation on a Unix platform. First, all functions specific to Unix or DOS are separated by an externally defined constant named UNIX. If the code is compiled on the Unix platform this variable is defined, otherwise it is not. For example, the following is a section of code from shell.c:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#ifndef UNIX
  #include <direct.h>
  #include <process.h>
  #include <graph.h>
  #include <dos.h>
#endif
```

In this section of code, the first four header files (those files with a .h extension) exist in both the DOS and Unix based libraries of the compilers used. The last four header files, however, exist only in the DOS compiler's libraries. These last four header files are thus separated by the #ifndef UNIX line of code. This line of code says that if the variable UNIX is not defined then compile the following commands until a #endif is encountered. Similarly, a #ifdef UNIX is also used throughout the models and routines to compile commands only when UNIX is defined. The UNIX variable is defined in make files, which are discussed below.

The second design consideration was to use *make* files to compile the code on the workstation. *Make* files are command generators which use description files to issue a sequence of Unix commands. They are typically used to sort out relationships between different files (Oram and Talbot, 1991). HTFS has 39 C programs, 22 header files, and 5 FORTRAN programs. Most of these programs have specialized functions that are accessed by other programs. Making one program's functions accessible to another program is accomplished by linking the programs during compilation. The management of these links is quite complex, so make files are used to manage the programs' relationships. HTFS's make files include the compiler and compiling options, define external variables, and link files. When a change is made in one routine, it is only necessary to remake the one file, and the make file will tell the compiler what other routines need to be relinked with the updated routine.

These two design considerations facilitated the initial compilation of HTFS's routines. What was necessary was to add two functions in utility.c to retrieve the path to

HTFS's sub-directories as well as the path to a temporary directory. Two environment variables are required to be set in order for the new functions to retrieve this information. The two environment variables required are named HTFSDIR and HTFSTMPDIR and should be set in the user's .cshrc file. A .cshrc file is a file which is read by the operating system when a user initially logs onto the system, and contains information about the users's environment (i.e., paths to executable code). The HTFSDIR environment variable is set to the path where all of HTFS's sub-directories are located, thus allowing HTFS's routines to find the directories from which it reads and writes files. The HTFSTMPDIR environment variable tells HTFS where some temporary files written by the GIS are located.

4.1.4 Testing and Debugging Outside the GIS

HTFS comes with four demonstration scenarios. The first scenario (htfsdmo1) is a basecase scenario, which is explained in detail in the following chapter, and the remaining scenarios are modifications of the basecase. These scenarios were used to test the code after it was complied on the Unix workstation. The scenario files were in ASCII format and the data in them can be processed by HTFS independently of the GIS. One file, config.tmp, contains information about the scenario to run. The path and scenario name were entered into this file, and then HTFS's main control routine (shell.c) was executed. All routines seemed to be executing correctly except for two: the path model, and the iterative proportional fitting (IPF) routine.

The path model returned an operating system error because a mathematical equation was attempting to raise a variable to the power of zero. It was assumed that some sort of data corruption occurred in the network file when transferring the files from the PC to the workstation. Therefore, a number of new networks were created using HTFS's netbuild utility, but all the new networks resulted in the same error. An undocumented change to HTFS's network structure was discovered upon further examination. Some of HTFS's routines were modified to read in additional variables from the network file. For example, the path model was modified to use an impedance function different from what was documented. Originally, the impedance was calculated based on speed and distance. The new impedance function includes speed, distance, access permissions, pavement type, median type, rural/urban code, and link restrictions (open or closed). In the network HTFS traditionally uses (a modified version of the NHPN Version 1.0), this information was not available. Therefore, a character item representing the new impedance items was added to the network and populated with zeros (0) signifying no data are available. The path model was then tested with the new information, and no errors were returned.

The second problem encountered was with the IPF routine. A temporary file named ipf.input is written to the directory from where HTFS is invoked. When the IPF routine is run, it is looking for the ipf.input file in the directory where the binary code is located. For this reason, whenever any of HTFS's routines are run, they need to be executed from the directory where the binary code is located, not from the directory from which it is invoked.

4.1.5 Reproducing TransCAD's Data Formats and Data Flow

Since HTFS was designed to run under TransCAD, it was engineered to take advantage of the method TransCAD uses to pass information to user routines. Additionally, HTFS has a specific routine to extract data directly from TransCAD's data tables. As discussed in Chapter 2, TransCAD passes information to user routines by writing out a solution.tmp file. The solution.tmp file written when HTFS is being run contains information about the ASCII scenario files to use, TransCAD data tables to use, and whether or not to extract coverages to ASCII. The main HTFS routine which interprets this file is loader.c; however, if the variable UNIX is defined, meaning the models are being executed on a Unix workstation, loader.c will not be executed. Since loader.c is used to extract and format data directly from TransCAD, its functions were reproduced using AML. The loader.c routine uses db_Vista libraries, as explained in Chapter 2, to read data directly from TransCAD's data tables, and then writes out space-separated columnar formated ASCII files of the data. An AML routine was written (extract.aml, Appendix C) to take the place of the extraction routines in loader.c. This unloads the data tables from Tables into an ASCII file. Tables, however, does not have an option to unload a file in a space-separated format. Therefore, a temporary item which is null (i.e., a blank space) was added to the data tables and is unloaded between items, creating a space-separated columnar formated ASCII file. Other options could be used, such as using cursors and the AML *write* function to write out the data, however, the method of unloading an individual null item was deemed to be more efficient and required no extra data processing.

One other HTFS routine, netbuild.c, also reads TransCAD's solution.tmp file. When netbuild is executed, it reads the solution.tmp file to retrieve the user-defined description of the network, and the network name. From TransCAD, the network is dumped to ASCII in a comma delimited format to a file named links.tmp. The first line of the file contains the name of the attributes which were dumped. When Tables unloads a file, it does not dump the name of the data table items, therefore the line of code in netbuild.c which strips the item information off was inactivated. Since this file was dumped from TransCAD in comma delimited format, the same is done from Arc/Info and no further processing is required.

Two report files also required minor modifications. The linkload.c and post.c routines write out report files suitable for import into TransCAD. The linkload.c routine prints out a summary file (linktrip.txt) of the number of trips occurring on each link, and a summary file (linktons.txt) of the number of tons traveling on a link. The post.c routine prints out a summary file (vmt.txt) of the annual vehicle miles of travel (VMT) per state, and a summary file (ptm.txt) of the annual payload ton miles (PTM) per state. In order to relate and import these files back into a TransCAD data table, TransCAD requires that the item names be listed on the first line of these files, Info, however, will not import a file with header information. Therefore, the lines of code in the linkload.c and post.c that print the header information were modified to redirect the item headers to new files with a .hdr extension (i.e., linktrip.hdr). An AML (import.aml, Appendix C) was written to read these header files, set up an Info data table based on the item names, and import the data.

A final change in data was necessary because of the different ways Arc/Info and TransCAD handle null numeric values. In TransCAD, a null numeric value is externally represented and exported as "--" (two dashes). This external representation is not valid in Arc/Info. Often, null numeric values are represented in Arc/Info as simply a "0" (zero). This however, is poor practice because null values can be mistaken as the true value of zero. Furthermore, zero is a valid value in HTFS's data tables. For example, if a link was shut down because of a bridge collapse, the maximum weight on that link can be set to zero, indicating the link is closed, and a scenario run to see how traffic shifts on the network. HTFS's netbuild.c routines and loader.c routines, look for data values equal to "--" and default them to -1, thus representing them as null or unknown. Since a value of "--" is not valid in an Arc/Info numeric field, all null values were defaulted to -1 in HTFS's data tables after they were imported to Arc/Info.

4.2 DESIGN AND DEVELOPMENT OF AML CODE

Arc/Info and TransCAD have very different methods for accessing user routines. TransCAD is a tightly-structured system with limited, but efficient, routines for passing information to a user's procedures. Arc/Info, conversely, is a very open system where the user has many options for passing information to a user's procedures when utilizing AML. Arc/Info's open system is both a strength and a weakness. It is a strength because it is fully customizable; however, a fairly large amount of AML code is required to integrate HTFS and Arc/Info, as compared to the amount of procedural code used to integrate HTFS with TransCAD.

The remainder of this chapter will focus on the design and development of the AML code created to integrate HTFS and Arc/Info. First, the design considerations will be discussed, followed by the development of major AML routines. For the details of the AML code, the interested reader is referred to Appendices C and D, and the menus developed are shown in Appendix E.

4.2.1 Design Considerations

There were two main design considerations, both of which were aimed at making HTFS user-friendly. First, a primary design consideration was the design of the Graphical User Interface (GUI). To facilitate previous HTFS users adopting the Arc/Info version of HTFS, the GUI was designed to be similar to HTFS's TransCAD GUI. Thus, a flow chart, similar to Figure 4-4, was drafted showing the logical flow of the original interface. This flow chart became the basis for not only the GUI, but also the AML code required to control the interface.

Another consideration in integrating HTFS with Arc/Info was the modules from which HTFS could be invoked. Originally, HTFS was going to be invoked from NETSHELL; however, it was decided that it would be more advantageous if the user could invoke HTFS from any of Arc/Info's modules. For example, some scenarios may require topological editing, which is performed in Arcedit or TED; some scenarios may require simple data table updates, which are easily performed in Tables; and some scenarios may require complex selection routines and in-depth analysis routines, which may be facilitated by using NETSHELL, which runs in Arcplot. Therefore, HTFS was

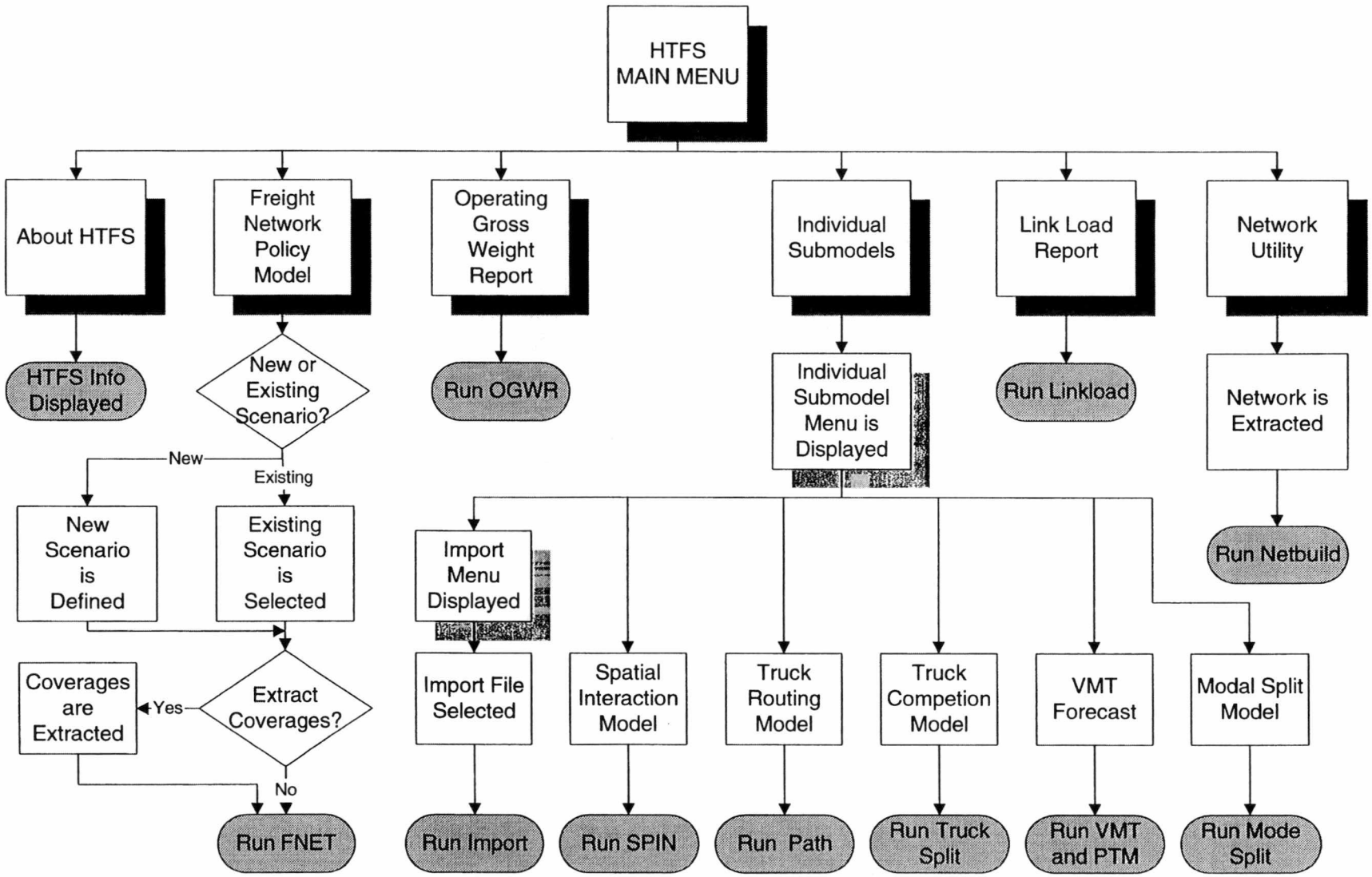


Figure 4-4 Flowchart of HTFS's GUI

designed to be executed from any of these modules, and easily integrated into existing AML software, such as NETSHELL or TED.

4.2.2 The Graphical User Interface

The GUI consists of a number of menus and sub-menus which display and/or prompt the user to input information. The menus consist of a number of different graphical objects called widgets, which are used for retrieving information from the user. The widgets used to create the GUI include the following:

check box widget: is used to set a Boolean variable to .TRUE. or .FALSE.;

choice widget: displays a list of choices in a scrolling menu from which the user can select one;

display widget: displays text or a variable's value as read-only text in a menu;

slider widget: displays a horizontal bar with a handle which can be slid along the bar to set the value of an integer;

text input widget: is used to provide a blank line where the user can type in input;

button widget: is used to execute commands using the variables set by other widgets (ESRI, 1993).

Figure 4-5 shows two different HTFS menus and how the widgets are displayed in the menus.

Based on the selection(s) made by the user from the menus, a sub-menu may be displayed requesting additional information. As the user is entering data to the menus, the information is being stored in variables. When all the information required from the user is gathered, the variables are passed to an AML, and the task the user selected is

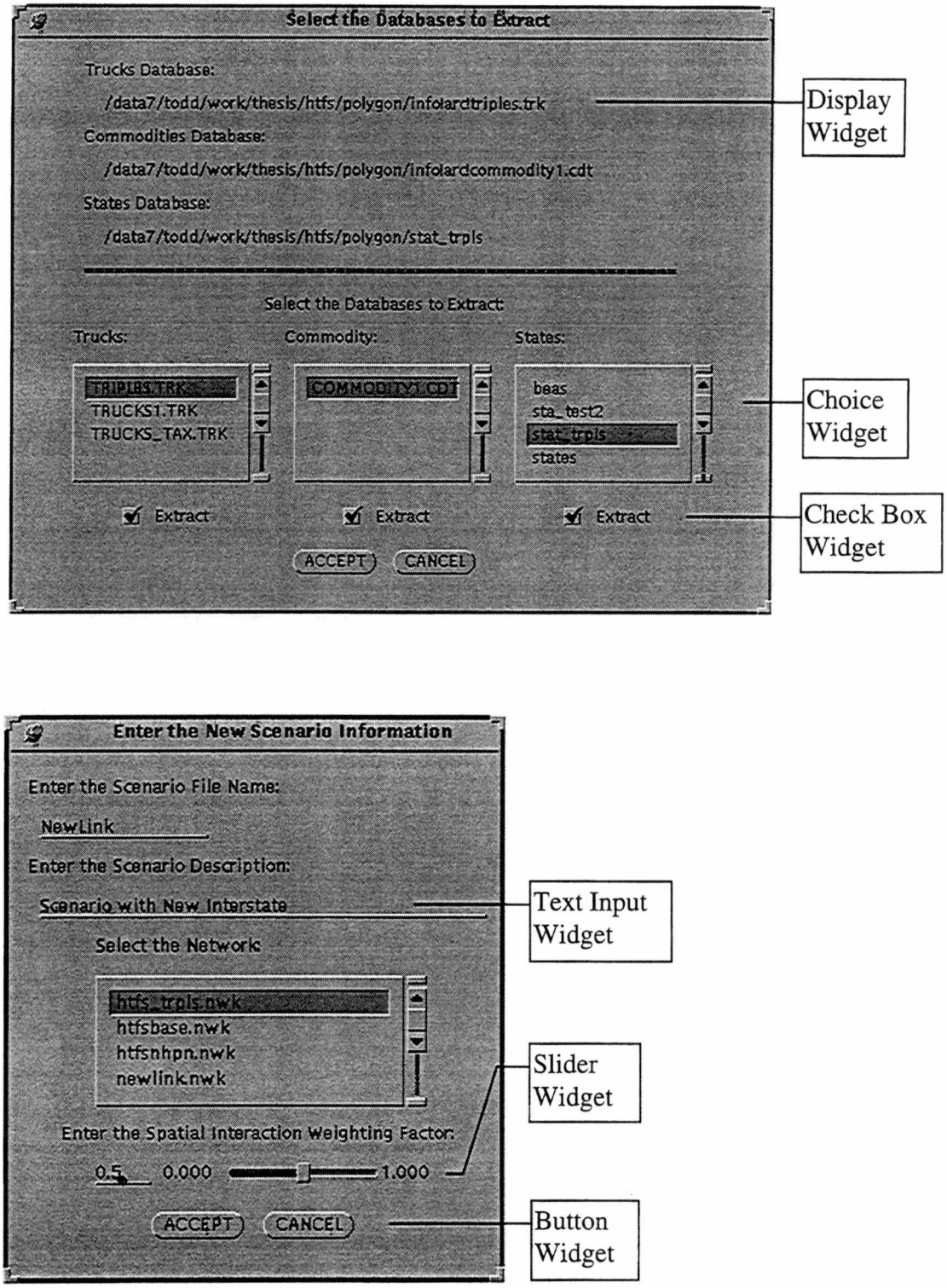


Figure 4-5: AML Menu Widgets

performed. Referring back to Figure 4-4, the flow chart of HTFS's GUI, the sub-menus which are displayed, and the operation which is performed can be seen.

4.2.3 Controlling the GUI

AML routines are used to control the flow of the menus, pass data to the models, and execute routines. There are two primary AMLs responsible for controlling the GUI, htfs.aml and fnet.aml, both of which will be discussed in more detail below.

HTFS.AML

The htfs.aml is the first AML executed when starting HTFS. The htfs.aml routine is responsible for displaying the main HTFS menu (htfs.menu), interpreting the selection made by the user, and then executing the appropriate operation. To display the main menu, htfs.aml creates a thread of the menu and then focuses control on the thread so that the AML processor knows input will be coming from this menu. Focusing the thread also causes the AML to halt until focus is returned to itself. When a selection is made from the main menu, a variable (named module) is set identifying which routine is to be executed, the menu is dismissed, and focus is returned to the AML. The AML then evaluates the variable to determine which routine to execute. For example, Figure 4-6 shows the main menu displayed when HTFS is executed from Arc/Info, a variable which is set when a selection is made, and how htfs.aml interprets the variable and executes the appropriate selection. The AML uses a &select directive (similar to a *select* statement in C) to select the contents of the variable and see which option it matches. For example, in

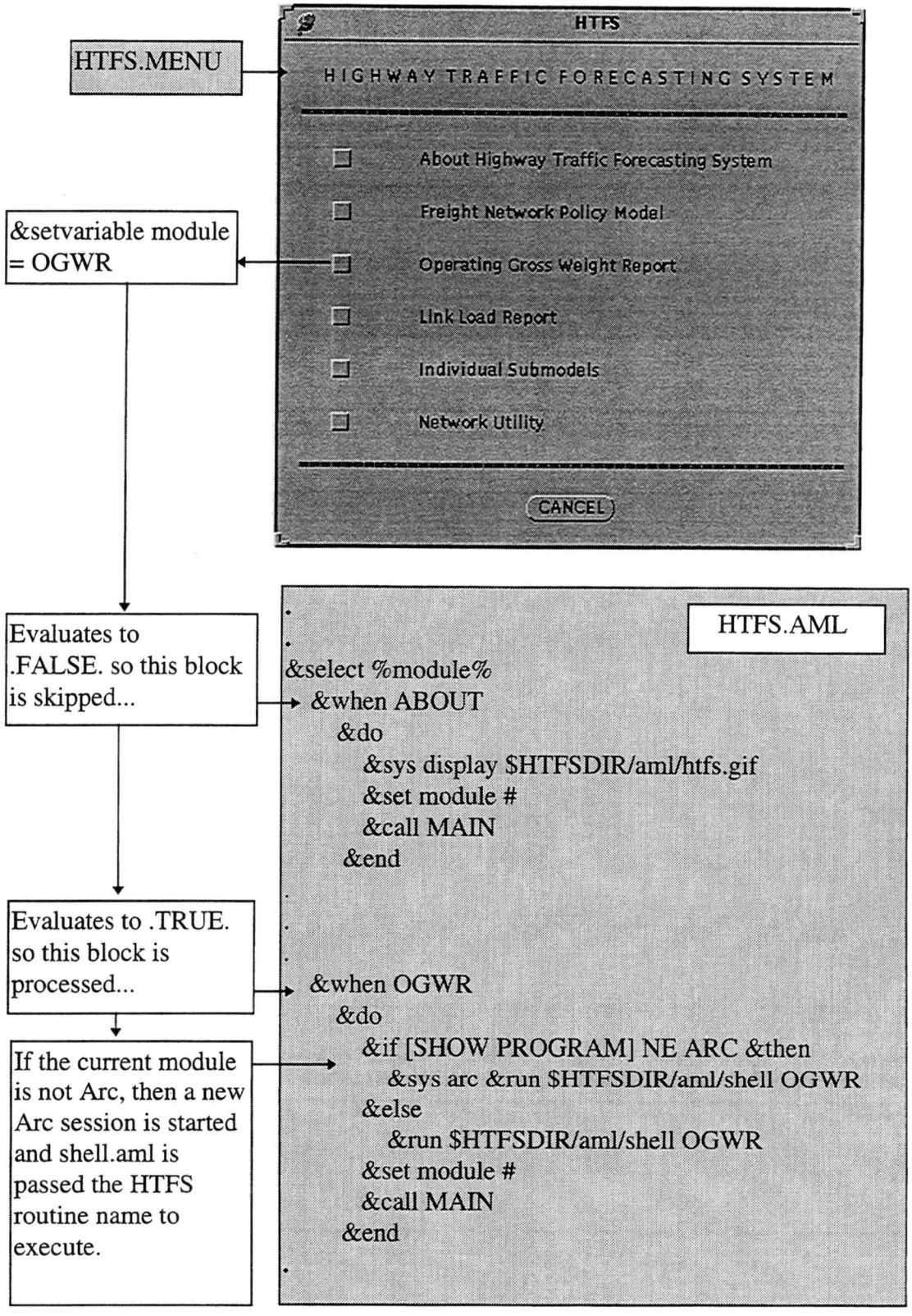


Figure 4-6
How HTFS.AML Interprets HTFS.MENU Selections and Executes Routines

the example given in Figure 4-6, the variable *module* is set to OGWR, thus the &when OGWR block is processed and HTFS's Operating Gross Weight Report routine is executed. After the selection has been executed, the main menu is redisplayed, and the user can make a new selection. The routine works similarly for the other options in the htfs.menu, except that some routines may require additional information, thus additional menus will be displayed to retrieve that information. However, when the Freight Network Policy Model is chosen, another control AML is executed which is described below.

FNET.AML

Fnet.aml is essentially the heart of the interface between Arc/Info and HTFS as it is responsible for defining scenarios, extracting data, writing files, and running HTFS. All of its menus are displayed in the same manner as htfs.aml, but the menus displayed by this AML are primarily used to retrieve the information necessary to create or retrieve and run a scenario. This information includes data table or scenario name(s), ASCII file names, weighting factors for the spatial interaction model, and whether or not to extract data tables. The code in fnet.aml is almost entirely broken into functions (represented as &routine in AML). In this AML, functions were advantageous because a number of routines need to be executed at different stages, depending on the user's selections. A &call directive (followed by the routine name) can be issued at any time in the AML, and the function's commands will be executed.

The fnet.aml also reproduces, or calls other AMLs which reproduce many of the functions that HTFS's loader.c and intrface.c performed under TransCAD. These routines include extracting Info data tables to ASCII, writing scenario files, and writing a config.tmp file.

4.2.4 Integrating Routines

A number of ASCII files needed to be produced which, when running under DOS, HTFS's loader.c, intrface.c, and TransCAD wrote. The AML routines developed to do this are instrumental in creating the integration of HTFS and Arc/Info. The routines which perform these tasks are fnet.aml, extract.aml, and network.aml.

The loader.c program was responsible for writing the scenario description file and extracting the states, commodity, and truck data tables from TransCAD. The scenario file contains all the information necessary to run a scenario, such as data table names, file names, and the spatial interaction scaling factor. Since the loader.c program is no longer called, the fnet.aml routine creates the scenario file from the information the user entered.

Figure 4-7 shows an example of a scenario file written by fnet.aml. When the user chooses to create a new scenario, the fnet.aml program displays menus which provide the user a choice, for each category (e.g., networks, states, commodity), of the available files which can be selected to define a scenario. Each of the selected choices are saved in variables, and after the user has input all the information required to define the scenario, the scenario file is written to the disk using AML's write function. When the user chooses to run the scenario, the scenario name is written to the config.tmp file by

NEWLINK.SNO

```
New Interstate from Phoenix to Denver...
29 Mar 95 16:51:57 Wednesday
/data7/todd/work/thesis/htfs/srf/newlink.nwk
/data7/todd/work/thesis/htfs/srf/supply.srf
/data7/todd/work/thesis/htfs/srf/demand.srf
/data7/todd/work/thesis/htfs/srf/opwtbase.ogw
/data7/todd/work/thesis/htfs/srf/railbase.srf
/data7/todd/work/thesis/htfs/srf/factdmo1.txt
htfsdmo1.trk
newlink.sta
htfsdmo1.cdt
0.5
181
0
1
2
3
4
5
.
.
.
177
178
179
180
```

Figure 4-7: Example of a Scenario File Written by FNET.AML

fnet.aml, and HTFS is invoked. HTFS's intrface.c program then retrieves the scenario name from config.tmp, and reads the scenario file. All the routines which are linked to intrface.c have access to this information also; therefore, HTFS has all the information necessary to model the scenario.

If any changes have been made to the States, Truck, or Commodity data tables, they need to be extracted to ASCII files before the models can process the new information. The loader.c program originally extracted the data directly from TransCAD, so an AML routine was written to perform this extraction from Info files. The fnet.aml queries users as to whether or not they want to extract any data tables. If the user chooses to extract a data table, a menu is displayed listing the states, trucks, and commodity data tables available for extraction. After the user selects the data table(s) to extract, the data table name(s) are passed to extract.aml. This routine changes directories to where the data tables are located, invokes Tables, and unloads the data tables to a columnar ASCII file which can be read by the models. Figure 4-8 shows a flow diagram of the Freight Network Policy Option and where the files are written or extracted to ASCII. The shaded boxes in this diagram show where primary changes occurred between the Arc/Info and TransCAD versions of HTFS.

If the Network Utility is selected from the main HTFS menu, then htfs.aml displays a menu which prompts the user to: 1) select the name of the network to extract; 2) enter a description of the network; and 3) choose the spatial interactions model's scaling factor. This information is then passed to network.aml which changes directories to where the network data table is located, invokes Tables, and unloads the network to an

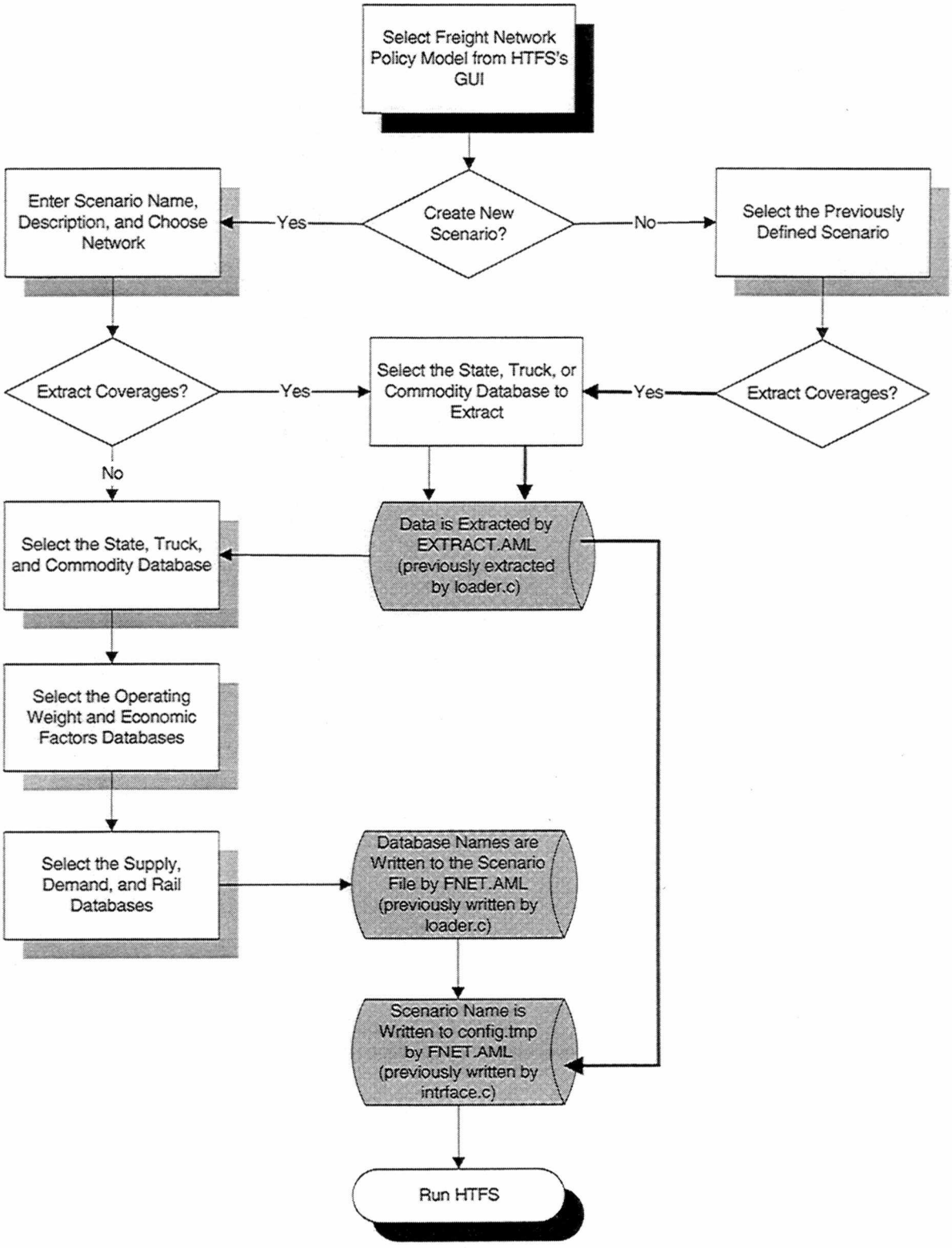


Figure 4-8: Flow of HTFS's Freight Network Policy Model Option, and When Information is Written to ASCII Files

ASCII comma delimited file. Next, the AML writes the name of the network, the network description, and the scaling factor to the solution.tmp file, which netbuild.c reads. Then the AML executes netbuild.c which converts the network to a forward star network.

An additional AML routine was also created which was not previously available in TransCAD. The import.aml routine imports the report files, output by HTFS's report generators, into Info files, which can then be analyzed. To import these files it was first necessary to redirect the header of the original report file, as described earlier in the chapter, to new files with a .hdr extension. The import.aml routine is passed the file type to import (ptm.txt, vmt.txt, linktons.txt, or linktrip.txt), and based on the file type passed, it reads the file's header file, defines the data table based on the header file, and imports the report file into the data table.

4.2.5 Executing HTFS

HTFS's Arc/Info GUI is started from any of Arc/Info's modules by typing &run htfs. This will display the main HTFS menu from which the user can select among its various options. When the user chooses to run an HTFS routine, the AML which is currently controlling the GUI checks to see if it is running in the Arc module or another Arc/Info module. If the user is in a module other than Arc, then a new Arc/Info session is started with the "&system arc" command, and the shell.aml routine, which executes HTFS's C programs, is executed. By starting a new Arc/Info session, any controls, such as threads, which may be running are effectively bypassed, and it is only necessary to use

the command syntax of the Arc module. If the current module is Arc, then the shell.aml is executed from the current session.

Since HTFS needs to be executed from the directory where the binary code is located, an AML routine named shell.aml was written which controls the execution of HTFS's routines. Each time one of HTFS's C programs is to be executed, the name of the routine to execute, is passed to shell.aml. This AML then changes to the directory where the binary code is located, and passes the routine name to the shell.c program, which controls the execution of HTFS's routines. Depending on the options the user has selected, htfs.aml, fnet.aml, or network.aml pass the routine name to shell.aml. After HTFS's routine has finished, the directory is changed back to the original directory, and control is returned to the HTFS interface.

HTFS's routines can be executed in succession or individually. HTFS's shell.c program controls how the routines are executed. This is done by passing the name of the routine to execute to shell.c, or passing the word MAIN to shell.c. If shell.c receives the name of a routine to execute, it only executes the one routine. If, however, MAIN is passed to shell.c, then all of FNET's modeling routines are executed in succession.

In the next chapter, three scenarios will be defined, executed, and run to demonstrate how HTFS is used in Arc/Info. The scenarios will be run from three different Arc/Info modules, and show some different methods for analyzing the model results.

CHAPTER 5

IMPLEMENTATION OF HTFS

To test the implementation and give a demonstration of HTFS in Arc/Info, three scenarios were designed, run, and then compared to a fourth, basecase, scenario. The first scenario examines an operational policy change to a law restricting the use of certain truck classes. The second scenario examines the impact of a structural change to the nation's highway network (i. e., a new road is built). The third scenario examines a pricing policy change in the form of a user tax. The scenarios were created from HTFS's original basecase scenario, and the results of the changes in vehicle miles of travel (VMT) and payload ton miles (PTM) were compared with the original basecase data. The basecase assumes no changes in policies, pricing, or network structure.

This chapter begins with a description of the basecase scenario, followed by a discussion of each of the scenarios, how they were created, and an analysis of their results. The analysis presented in this chapter is not a comprehensive evaluation, as there are many parameters with complex interactions involved in understanding how freight is shipped throughout the country. Rather, this analysis is meant as a demonstration of some different methods for visualizing the results within the GIS.

5.1 THE BASECASE SCENARIO

In its original DOS version, HTFS's scenarios were created by establishing a number of TransCAD data tables with default data and parameter values. Ray (1992) described how the default values included allowing all trucks, excluding doubles and

triples, to operate in all contiguous states, and the trucks were given a maximum operating gross vehicle weight limit (OGW) of 80,000 pounds. The remaining data and data tables were populated from a variety of different sources (Appendix F). HTFS's basecase scenario (htfsdmo1) was created by further modifying this initial data set. The States data table was updated with more accurate OGW data for each state, and the Truck data table, likewise, was updated with more accurate OGW data for the 14 different truck types, according to the 1987 Truck Inventory and Use Survey (TIUS 1987). The States data table was then updated to more accurately reflect the permissions, or allowability, of different truck configurations operating on each state's network. For example, some states were updated to allow double- and triple-trailer trucks to operate on their networks. No data about link weight limits were updated in the network data table, so these data are taken directly from the States data table, which contains default values (Ray 1992).

The basecase scenario models the flows of 23 different commodities, which are shown in Appendix G-1, between 181 BEA regions. An example of the basecase supply and demand for farm, fish, and marine products by BEA region is shown in Appendix G-2. Based on these flows, HTFS reports the total VMT and PTM for each state, as well as the total VMT and PTM for each link, on an annual basis.

5.2 SCENARIO 1: Allowing Triples on All Interstates

Overview

A number of states allow longer combination vehicles (LCVs) to operate within their borders. According to David Middendorf (1995) of ORNL's Center for Transportation Analysis:

> Thirteen states allow Rocky Mountain doubles, comprised of a tractor pulling a 45- to 48-foot trailer followed by a 28-foot "pup." Nine states permit turnpike doubles consisting of a tractor pulling two 48-foot trailers, and ten western states allow double- and triple-trailer configurations.

The trucking industry is in favor of using LCVs because they have a greater potential for productivity (Middendorf 1995). Many groups, however, are opposed to LCVs, especially triples, because of safety and other concerns. Thus, only a small number of states have legalized triples. This first scenario was designed to evaluate what the impact on PTM and VMT would be if all contiguous allowed triples on their Interstates.

Defining the Scenario and Running HTFS

To allow triples on all Interstates, it was necessary to update two data tables. First, HTFS's basecase network and States data tables were copied to new coverages so the basecase data would not be affected. NETSHELL was used to select all Interstates, and update the network data table to allow triples to operate on these links. NETSHELL was then used to update the States data table to allow triples to operate in all states. Updating the States data table was necessary because its permissions have priority over the network permissions. After these changes were made, the scenario was run.

HTFS's GUI was started from NETSHELL by selecting "HTFS" from its main menu. From the HTFS menu, the "Network Utility" option was chosen, and the network previously updated was selected and extracted to ASCII. After the network was extracted, the "Freight Network Policy Module" option was selected, and the scenario was defined in the same manner as was outlined in Figure 4-6. The new scenario's name (triples.sno) and a description of the scenario were entered. The States data table was selected and extracted to ASCII, and the basecase's Truck and Commodity data tables were chosen to include in the scenario. The basecase's Operating Gross Weight, Economic Factors, Supply, Demand, and Rail Cost data tables were also used in this scenario. After all the data tables were selected, the scenario was run, and the results were imported into Arc/Info by selecting the "Import Text Files" option from HTFS's GUI.

Evaluating the Results

Since it is more productive for trucking companies to use triples, it was expected that allowing their use in all contiguous would show a dramatic increase in their use. To visualize the impact of allowing their use, three maps were created. The first two maps, Figure 5-1 and 5-2, are flow maps of VMT before and after the scenario was run, respectively. These maps were generated in NETSHELL, and the reader should note that the legends, which associate the line thickness with VMT, are scaled differently on the



Figure 5-1: Basecase Flows of Triples on Interstates
(In Millions/Year)



Figure 5-2: Forecasted Flows of Triples on Interstates
(In Millions/Year)

two maps. This occurred because NETSHELL's flow routine scales the legend based on the minimum and maximum values in the results field, which changed.

Figure 5-1 shows the VMT of triples on the Interstates in the basecase scenario, and depicts the flows in millions of trips per year. As can be seen from this map, the widest use of triples is in the Rocky Mountain and Great Plains region, where they are allowed in a number of states. The map also shows some heavy use in Ohio and Indiana, as well as limited use in Florida, where they are also legal. The second flow map, Figure 5-2, shows the forecasted results with triples being allowed on all Interstates in all contiguous states. This map shows how all states receive traffic from triples after they are legalized. When compared to the basecase map, it can be seen that the links which originally had the largest percent of VMT, remain as some of the heaviest traveled links. This is especially true in Ohio and its surrounding states, where there is the greatest amount of triples in use. It would be interesting to investigate this area further to identify what commodities have a propensity to use triples when available. This could be done by modifying some of HTFS's report routines to report the flows by commodity group; however, this is beyond the scope of this study. Figure 5-2 also shows how the areas with the greatest population concentrations are the ones where triples are likely to be heavily used, if allowed. These flows can be seen on the map in Southern California, California's Bay area, and the North East Coastal area. This makes sense, as these areas are the destination for a large amount of freight, and, therefore, trucking companies could increase their efficiency and productivity if they could use triples to ship freight to these areas.

In addition to the flow maps, the shift in truck classes can be seen in Figure 5-3. This figure shows a comparison of the national totals of VMT and PTM of the basecase and triples scenario's. The PTM was not calculated for the first three truck classes because these truck classes were not included in the basecase scenario. The "Total Shift" row is the difference in VMT and PTM after the models were run, while the "Percent Shift" row is the percent change in VMT and PTM after the scenario was run. The "Forecasted" row is the national total of VMT and PTM after the scenario was run, and the "Basecase" row represents the national total of VMT and PTM in the basecase scenario. This figure shows how the increase in the use of triples results in the decrease in the use of the smaller traditional trucks. This seems likely because the increased use of triples means less freight would be available to be carried by the traditional trucks. However, the larger double's VMT and PTM increased moderately, thereby suggesting that allowing triples creates an overall shift to using LCVs.

5.3 SCENARIO 2: A New Interstate Between Denver and Phoenix

Overview

Topological changes can have a dramatic affect on the overall efficiency of a network. Changes to a network, such as building a new road, can dramatically reduce the amount of time required to get from one place to another; thereby increasing the interaction of people or goods between the two places. Therefore, it is common for a local topological change on a network to have a global impact.

		PTM	VMT
Single Unit 2 Axles (SU2)	Total Shift:	0	286.55
	Percent Shift:	0	2.34
	Forecasted:	0	12550.69
	Basecase:	0	12264.14
Single Unit 3 or 4 Axles (SU3&4)	Total Shift:	0	58.46
	Percent Shift:	0	2.29
	Forecasted:	0	2606.03
	Basecase:	0	2547.56
Conventional Semi 3 Axle (CS3)	Total Shift:	0	54.24
	Percent Shift:	0	2.46
	Forecasted:	0	2258.65
	Basecase:	0	2204.41
Conventional Semi 4 Axles (CS4)	Total Shift:	-1094.35	-164.72
	Percent Shift:	-2.8	-2.8
	Forecasted:	38028.49	5721.95
	Basecase:	39122.84	5886.67
Conventional Semi 5 Axles (CS5)	Total Shift:	-14453.3	-1237.36
	Percent Shift:	-2.1	-2.12
	Forecasted:	672883	57105.72
	Basecase:	687336.3	58343.07
Conventional Semi 6 Axles (CS6)	Total Shift:	-632.73	-52.26
	Percent Shift:	-2.92	-2.96
	Forecasted:	21016.53	1714.06
	Basecase:	21649.25	1766.32
Combined Tractor 4 Axles (CT4)	Total Shift:	-133.18	-15.62
	Percent Shift:	-3.02	-3.02
	Forecasted:	4273.49	501.08
	Basecase:	4406.68	516.7
Combined Tractor 5 Axles (CT5)	Total Shift:	-125.32	-8.74
	Percent Shift:	-2.96	-2.96
	Forecasted:	4108.92	286.36
	Basecase:	4234.24	295.1
Combined Tractor 6 Axles (CT6)	Total Shift:	-44.1	-2.88
	Percent Shift:	-3.06	-3.06
	Forecasted:	1396.66	91.21
	Basecase:	1440.75	94.09

LONG COMBINATION VEHICLES

Rocky Mountain Doubles (DS5)	Total Shift:	-916.38	-64.69
	Percent Shift:	-2.66	-2.69
	Forecasted:	33569.71	2339.11
	Basecase:	34486.1	2403.8
Western Doubles (DS7)	Total Shift:	10.68	0.67
	Percent Shift:	0.72	0.64
	Forecasted:	1488.83	105.44
	Basecase:	1478.14	104.77
Turnpike Doubles (DS9)	Total Shift:	8.22	0.25
	Percent Shift:	0.73	0.73
	Forecasted:	1130.02	33.81
	Basecase:	1121.79	33.57
Triples (TRP)	Total Shift:	5151.38	253.79
	Percent Shift:	1394.73	1292.88
	Forecasted:	5520.72	273.42
	Basecase:	369.35	19.63
User Defined Alternatives (Undefined Triple) (XXX)	Total Shift:	4655.63	228.63
	Percent Shift:	1357.2	1255.29
	Forecasted:	4998.67	246.85
	Basecase:	343.03	18.21

Figure 5-3: Triples Allowed on All Interstates
National VMT and PTM Totals

Building a new Interstate between Denver and Phoenix has been discussed by the Center for the New West, a special interest group, but never seriously enough to bring it to the proposal stage (Lewis 1995). Nevertheless, the idea of a new Interstate between the two cities, which would certainly come at a tremendous cost, poses some interesting questions as to what would happen to the VMT and PTM in the region, as well as nationally, if it were built. Therefore, a scenario was developed to examine the impact to VMT and PTM of building the Interstate.

Defining the Scenario and Running HTFS

According to Steve Lewis of the Department of Transportation's Bureau of Transportation Statistics (1995), the discussion of the Interstate never reached the stage of location or alignment; therefore a great deal of flexibility was available to locate the Interstate. The location of the new Interstate was chosen by examining the topography of the region in order to propose a feasible location. Attempting to minimize the mileage of new construction, the new Interstate was added between I70 in Utah, at the intersection with U191, to I17 in Arizona, at the intersection with I40. Figure 5-4 shows the location where the Interstate was added. To estimate the length of the new Interstate, a critical factor in determining the path truck freight will take, the average of the percent of error between the straight line links' distance and actual distance of the links in the region was determined. In other words, because the base network is a straight line network, each link has a percent of error between its straight line distance (the distance between the from- and to-node of the link) and its actual distance (the true observed distance of the link).

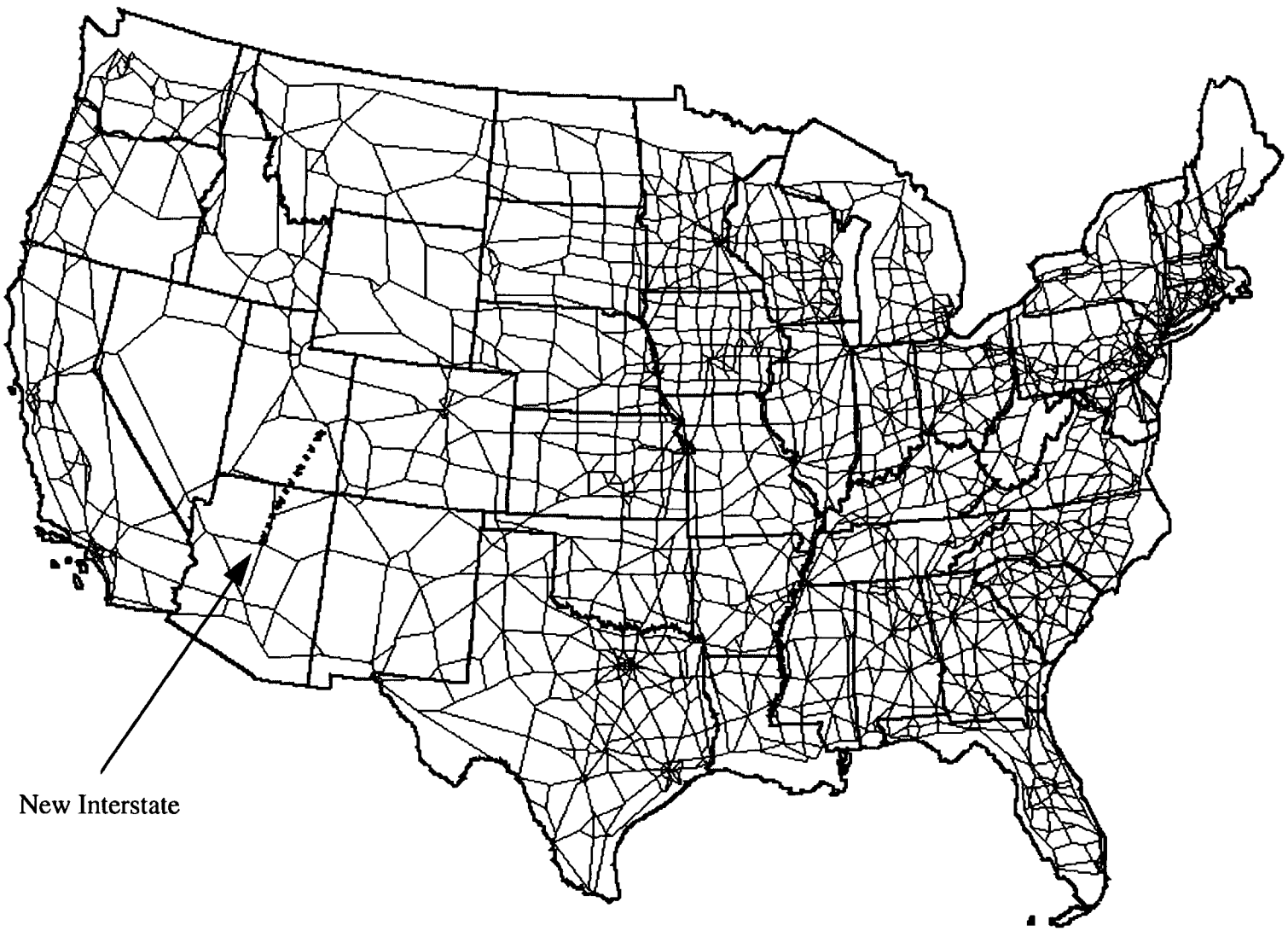


Figure 5-4: Location of New Interstate

The percent of error for the links immediately surrounding the new Interstate were calculated, and the average percent of error was determined. This average percent of error was then applied to the straight line distance of the new Interstate, and an estimate of the Interstate's length was able to be determined. The Topological EDiting System (TED) was used to add the new Interstate, to determine the surrounding links' straight-line and actual distances, and to update the Interstates attributes.

After the link was added, its attributes updated, and the information saved, HTFS was activated from TED's main menu. The same procedures were used to define this scenario as those described in Scenario 1, except that the new network was the only data table extracted to ASCII. All the other data tables used were from HTFS's basecase scenario.

Evaluating the Results

After the models were run, the results were imported into Info using HTFS's import utility. The area of particular interest for this scenario was the area surrounding the new Interstate, which included the links previously used for travel between Denver and Phoenix. These links were selected in TED and extracted to a new cover, allowing a much smaller network to be evaluated. A new item was added into the new coverage's AAT named "tot_trips" and was populated with the basecase total trips for the links. A "results" field was then populated with the total trips from the scenario's forecast. Next an item named "difference" was added and populated by subtracting the "tot_trips" from the "results" fields (i.e., difference = results - tot_trips). This field allowed the increase or

decrease on the links to be easily evaluated. A map, Figure 5-5, was then created in TED showing the links and the increase or decrease (in millions) of total annual trips on the links between Denver and Phoenix. The new Interstate's flow amount represents the actual flow along it, since no previous flow existed. The map clearly shows a dramatic impact on the number of trips occurring on the links in the region. Originally, a large percentage of the flow was occurring on highways between the two cities, as depicted in Figure 5-5; however, when the option of a new Interstate was presented, a dramatic decrease (as much as a 140 million trips annually) on some of these highways occurred.

Two flow diagrams, Figures 5-6 and 5-7, of the region were created using NETSHELL to help visualize the impact of VMT in the region. The user should note that the legends, which associate the line thickness with VMT, are scaled differently on the two maps. Figure 5-6 is a flow diagram of the basecase VMT between Denver and Phoenix. This figure shows how a large amount (over 317 million trips/year on some links) of freight traffic between the two cities travels on highways in a south-westerly to north-easterly direction. Comparing Figures 5-6 to 5-7, which is a flow diagram of VMT after the scenario was run, the shift of traffic from these highways to the new Interstate can be easily seen.

Two tables were created to analyze the impact on different truck classes at a local and national level. Figure 5-8 shows the aggregated change in VMT and PTM in Arizona, Colorado, New Mexico, and Utah. As expected, each of the traditional truck classes shows a modest increase in VMT and PTM because of the more efficient network (i.e., the networks capacity has increased and travel times through the region of the new

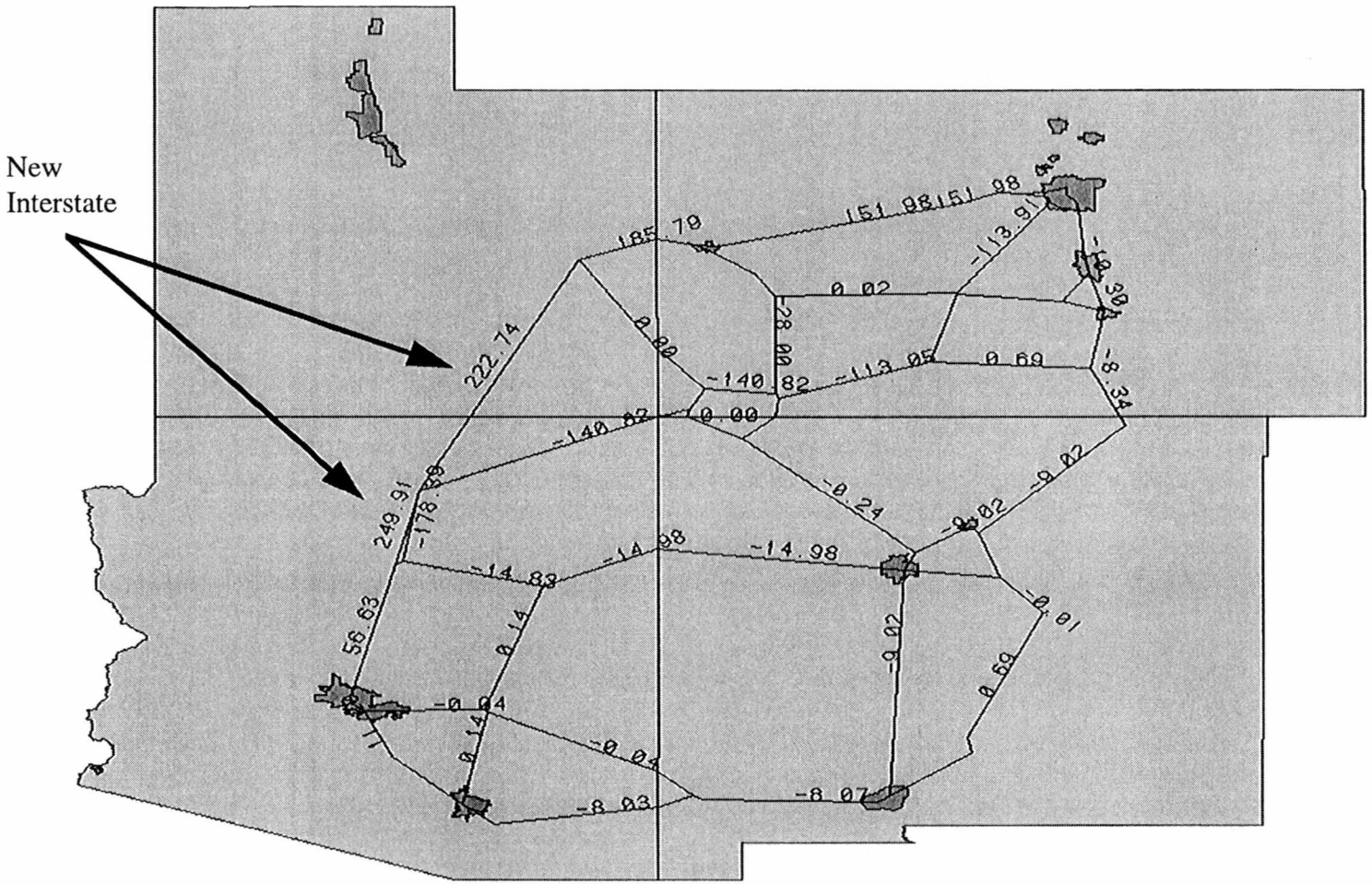


Figure 5-5: Increase or Decrease in VMT after New Interstate is Added
(In Millions/Year)

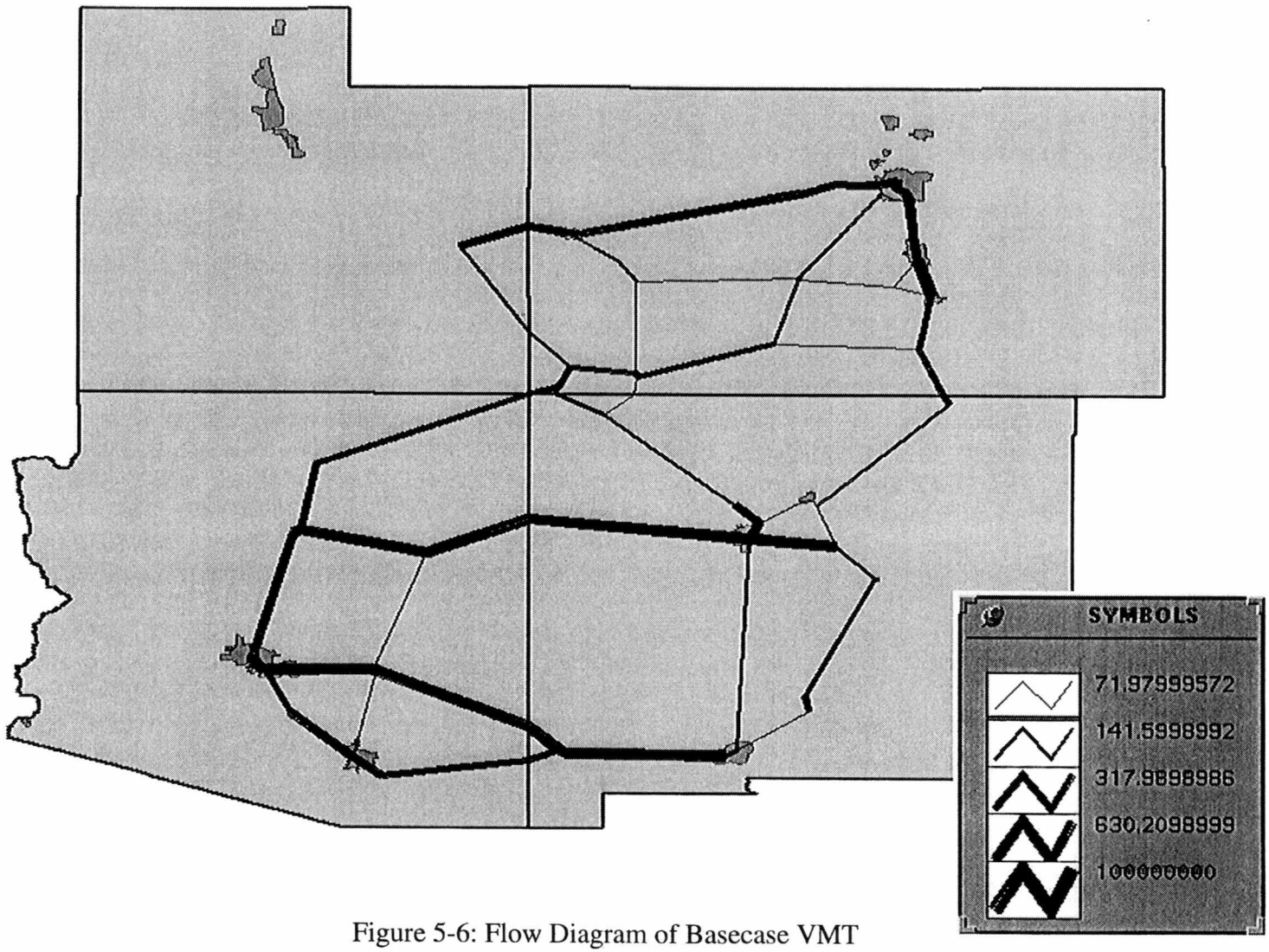


Figure 5-6: Flow Diagram of Basecase VMT
without Interstate (In Millions/Year)

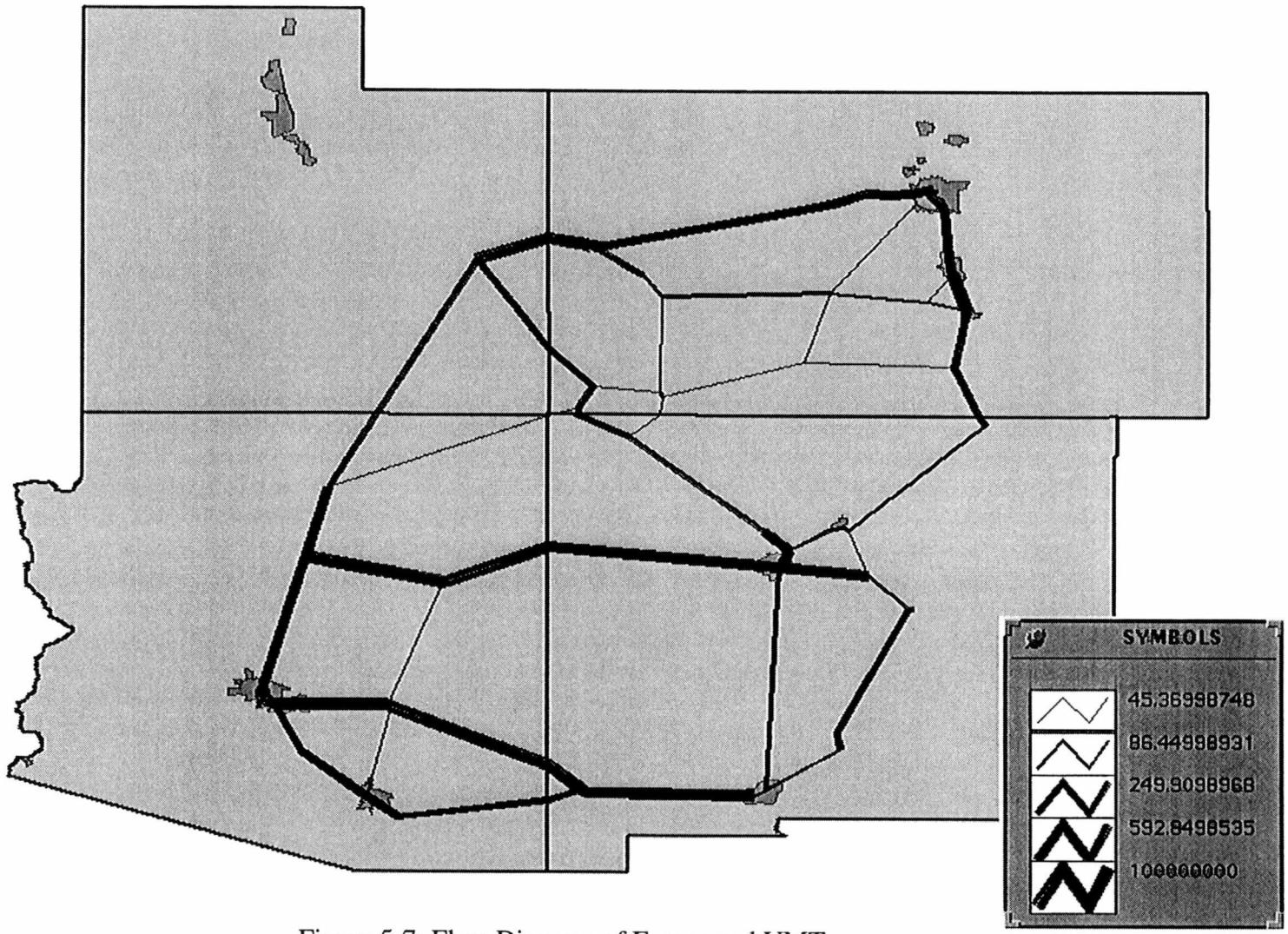


Figure 5-7: Flow Diagram of Forecasted VMT
with New Interstate (In Millions/Year)

		PTM	VMT
Single Unit 2 Axles (SU2)	Total Shift:	0	0.08
	Percent Shift:	0	0.02
	Forecasted:	0	500.01
	Basecase:	0	499.93
Single Unit 3 or 4 Axles (SU3&4)	Total Shift:	0	0.01
	Percent Shift:	0	0.01
	Forecasted:	0	91.76
	Basecase:	0	91.74
Conventional Semi 3 Axle (CS3)	Total Shift:	0	0.02
	Percent Shift:	0	0.02
	Forecasted:	0	94.81
	Basecase:	0	94.79
Conventional Semi 4 Axles (CS4)	Total Shift:	5.77	0.87
	Percent Shift:	0.63	0.63
	Forecasted:	920.68	138.53
	Basecase:	914.91	137.66
Conventional Semi 5 Axles (CS5)	Total Shift:	146.67	11.5
	Percent Shift:	0.36	0.33
	Forecasted:	41191.47	3494.23
	Basecase:	41044.8	3482.73
Conventional Semi 6 Axles (CS6)	Total Shift:	2.56	0.17
	Percent Shift:	0.59	0.47
	Forecasted:	438.55	36.62
	Basecase:	435.99	36.45
Combined Tractor 4 Axles (CT4)	Total Shift:	0.4	0.05
	Percent Shift:	0.5	0.5
	Forecasted:	80.68	9.46
	Basecase:	80.28	9.41
Combined Tractor 5 Axles (CT5)	Total Shift:	0.46	0.03
	Percent Shift:	0.55	0.55
	Forecasted:	82.97	5.78
	Basecase:	82.51	5.75
Combined Tractor 6 Axles (CT6)	Total Shift:	0.12	0.01
	Percent Shift:	0.46	0.47
	Forecasted:	25.55	1.67
	Basecase:	25.44	1.66

LONG COMBINATION VEHICLES

Rocky Mountain Doubles	Total Shift:	6.01	0.38
	Percent Shift:	0.62	0.57
	Forecasted:	975.64	67.5
	Basecase:	969.63	67.12
Western Doubles	Total Shift:	13.76	1.02
	Percent Shift:	8.64	9.74
	Forecasted:	172.96	11.48
	Basecase:	159.2	10.46
Turnpike Doubles	Total Shift:	11.18	0.33
	Percent Shift:	9.15	9.15
	Forecasted:	133.41	3.99
	Basecase:	122.23	3.66
Triples	Total Shift:	3.78	0.22
	Percent Shift:	7.9	8.9
	Forecasted:	51.66	2.75
	Basecase:	47.87	2.52
User Defined Alternatives (Undefined Triple)	Total Shift:	3.13	0.19
	Percent Shift:	7.13	8.06
	Forecasted:	47.05	2.49
	Basecase:	43.92	2.31

Figure 5-8: New Interstate Between Denver and Phoenix
VMT and PTM Totals for Arizona, Colorado, New Mexico, and Utah

Interstate have decreased); however, a large relative increase occurred in the LCVs (doubles and triples). This is most likely the result of triples being allowed in three of the four states (Arizona, Colorado, and Utah). This shift to LCVs is also apparent in Figure 5-9, which shows the aggregated changes in VMT and PTM on a national scale. Figure 5-9 shows a national decline, albeit modest, in the use of traditional truck classes. The decline in these truck classes, however, is made up for by an increase in the LCVs, indicating the more efficient use of LCVs, especially for long-haul trips, through this region. Looking on a larger scale, the increased traffic drawn to this region (indicated by the increases in VMT and PTM in Figure 5-8), is a good example of how a localized change can have global impacts.

5.4 SCENARIO 3: Increasing the fuel tax by 10 percent.

Overview

Changes in pricing policies are often proposed to raise funds from users of government funded projects. For example, it is common for the Federal government to raise fuel taxes to help offset the increasing costs of maintaining and building new roads. HTFS has the ability to model the impact of raising fuel taxes on VMT and PTM by adjusting the operating cost per mile of trucks to reflect the proposed increase. Therefore, a scenario was designed to determine the impact of raising the fuel tax by 10 percent.

		PTM	VMT
Single Unit 2 Axles (SU2)	Total Shift:	0	2.21
	Percent Shift:	0	0.02
	Forecasted:	0	12266.34
	Basecase:	0	12264.14
Single Unit 3 or 4 Axles (SU3&4)	Total Shift:	0	0.47
	Percent Shift:	0	0.02
	Forecasted:	0	2548.04
	Basecase:	0	2547.56
Conventional Semi 3 Axle (CS3)	Total Shift:	0	0.38
	Percent Shift:	0	0.02
	Forecasted:	0	2204.79
	Basecase:	0	2204.41
Conventional Semi 4 Axles (CS4)	Total Shift:	-0.27	-0.02
	Percent Shift:	0	0
	Forecasted:	39122.57	5886.65
	Basecase:	39122.84	5886.67
Conventional Semi 5 Axles (CS5)	Total Shift:	-109.18	-10.11
	Percent Shift:	-0.02	-0.02
	Forecasted:	687227.1	58332.96
	Basecase:	687336.3	58343.07
Conventional Semi 6 Axles (CS6)	Total Shift:	-0.32	-0.08
	Percent Shift:	0	0
	Forecasted:	21648.93	1766.24
	Basecase:	21649.25	1766.32
Combined Tractor 4 Axles (CT4)	Total Shift:	-0.13	-0.01
	Percent Shift:	0	0
	Forecasted:	4406.55	516.68
	Basecase:	4406.68	516.7
Combined Tractor 5 Axles (CT5)	Total Shift:	-0.09	-0.01
	Percent Shift:	0	0
	Forecasted:	4234.16	295.09
	Basecase:	4234.24	295.1
Combined Tractor 6 Axles (CT6)	Total Shift:	-0.05	0
	Percent Shift:	0	0
	Forecasted:	1440.7	94.09
	Basecase:	1440.75	94.09

LONG COMBINATION VEHICLES

Rocky Mountain Doubles (DS5)	Total Shift:	-0.44	-0.06
	Percent Shift:	0	0
	Forecasted:	34485.65	2403.74
	Basecase:	34486.1	2403.8
Western Doubles (DS7)	Total Shift:	14.46	1.07
	Percent Shift:	0.98	1.02
	Forecasted:	1492.61	105.85
	Basecase:	1478.14	104.77
Turnpike Doubles (DS9)	Total Shift:	11.79	0.35
	Percent Shift:	1.05	1.05
	Forecasted:	1133.58	33.92
	Basecase:	1121.79	33.57
Triples (TRP)	Total Shift:	3.95	0.23
	Percent Shift:	1.07	1.2
	Forecasted:	373.3	19.86
	Basecase:	369.35	19.63
User Defined Alternatives (Undefined Triple) (XXX)	Total Shift:	3.25	0.19
	Percent Shift:	0.95	1.06
	Forecasted:	346.28	18.41
	Basecase:	343.03	18.21

Figure 5-9: New Interstate Between Denver and Phoenix
National VMT and PTM Totals

Defining the Scenario and Running HTFS

For this scenario, the Truck data table was updated to reflect the fuel cost increase after it was copied to a new Info file. The cost-per-mile field in the Truck data table contains an estimate of the overall operating cost-per-mile for each truck class. The overall cost is a summation of the following variables: labor, vehicle, fuel, tires, repair, and overhead. Since the fuel tax only affects the fuel variable, a 10 percent increase in the fuel cost was determined by increasing the fuel variable by 10 percent. The product was then added to the cost-per-mile field in the truck's data table for each truck class. From the Arc module, HTFS was started by typing "&run htfs". A new scenario representing the increased operating cost was created in the same manner as described in Scenario 1, except that the modified Truck data table was selected and extracted to ASCII. The remaining data tables used were from the basecase scenario.

Evaluating the Results

After the models were run, the results were imported into Info using HTFS's import utility. Since overall decrease in VMT and PTM per state were of interest in this scenario, two maps, Figures 5-10 and 5-11, were created in TED depicting this information. Figure 5-10 shows the percent of decrease in VMT by state, and Figure 5-11 is a choropleth map representing the percent decrease in VMT. From Figure 5-11 it is apparent that the largest decreases in VMT occur in the Rocky Mountain and Great Plains region. These decreases are likely because trips in this part of the country tend to be

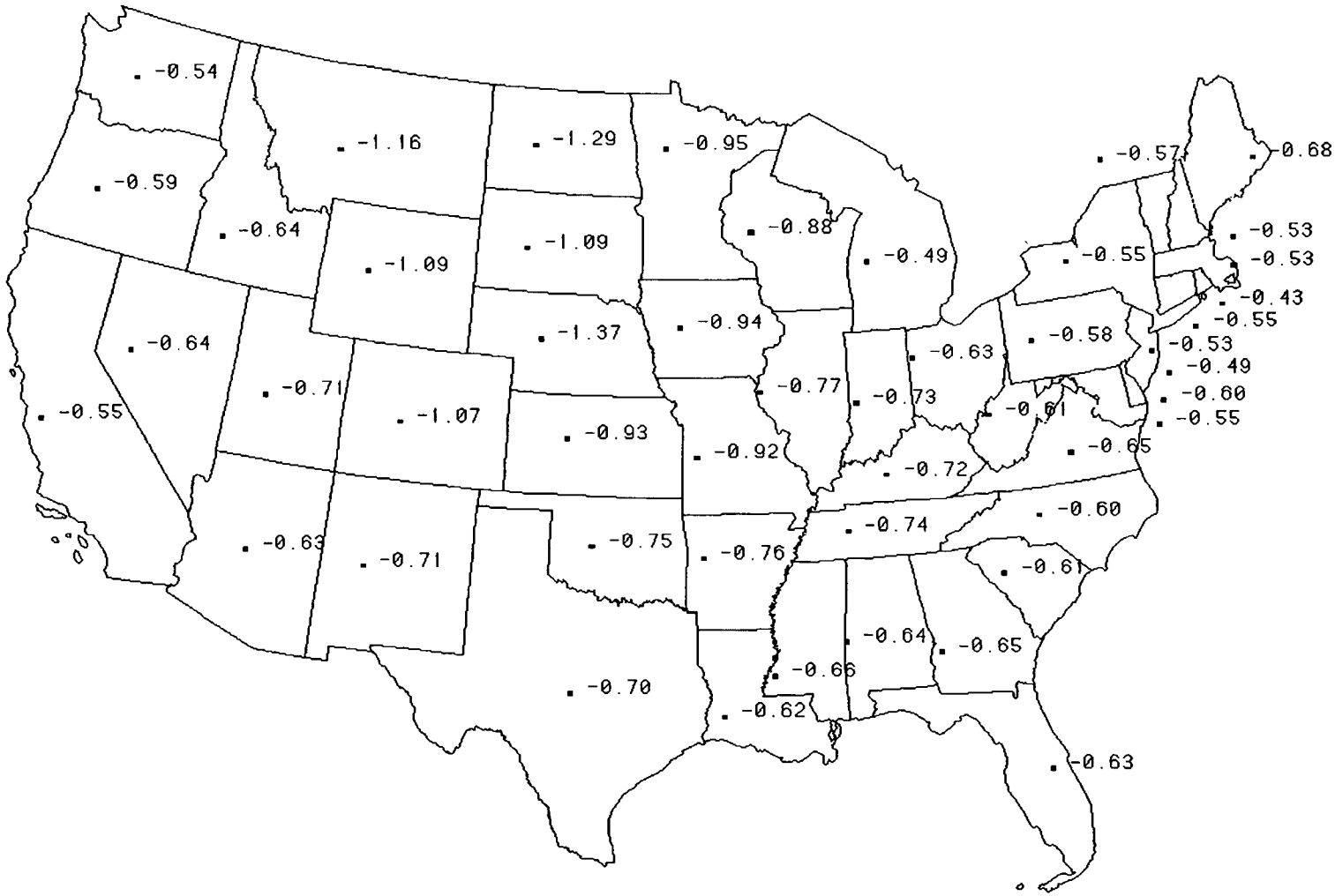


Figure 5-10: Percent Change in VMT after 10 Percent Fuel Tax

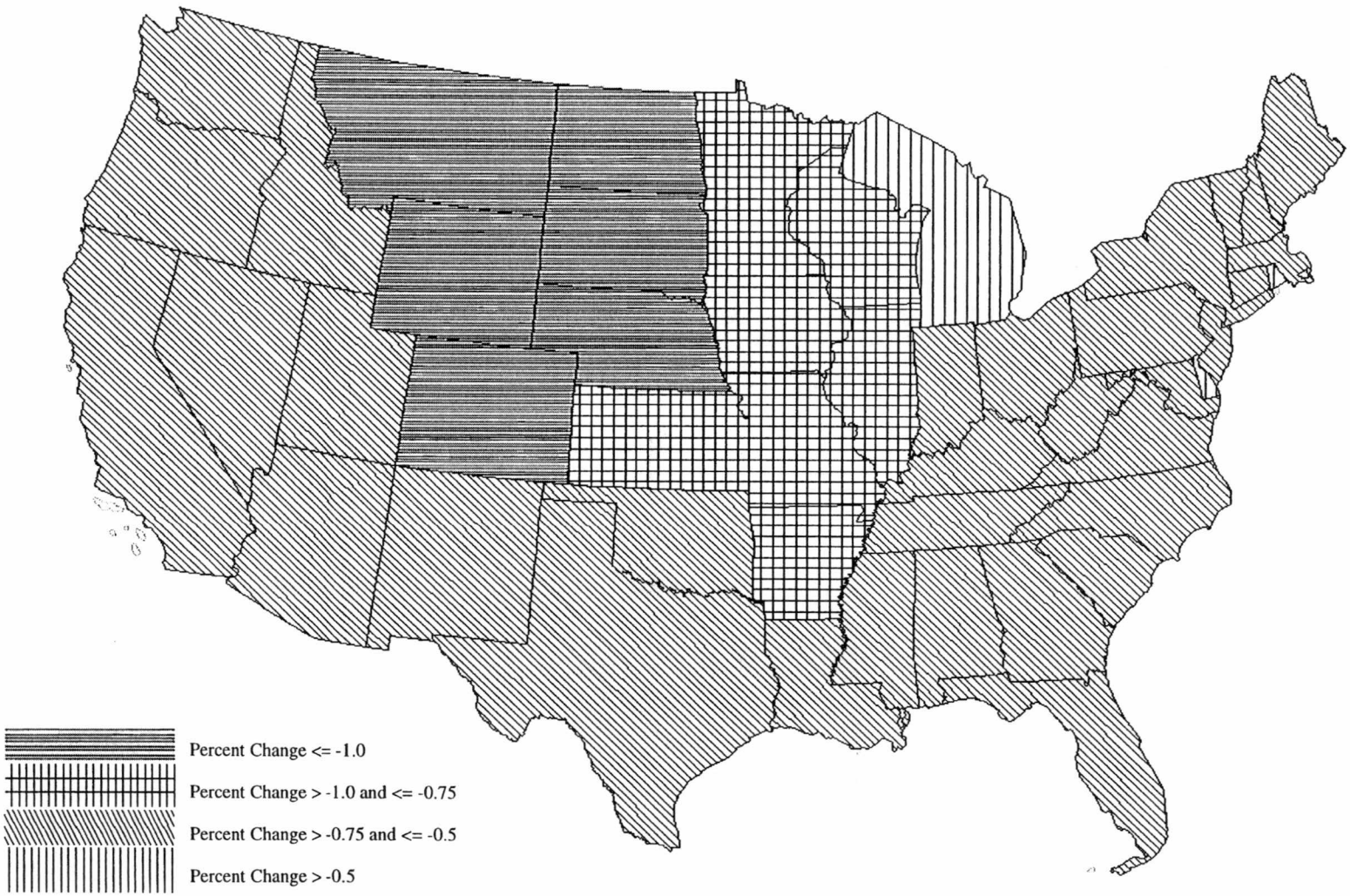


Figure 5-11: Percent Decrease in VMT after 10 Percent Fuel Tax

long-haul, thus they would be more affected by the increase in fuel prices. Figure 5-12 shows the national totals of VMT and PTM before and after the scenario was run. From this figure, it can be seen that the LCVs have a much greater decline in VMT then the other truck classes.

Summary

These three scenarios, along with the basecase, demonstrate how HTFS can be used within a GIS to define and analyze a scenario. The scenarios also demonstrate how HTFS can not only evaluate policy changes, such as allowing triples on all Interstates or increasing the fuel tax, but HTFS can also evaluate physical changes to the network, such as adding a new link. In addition, these scenarios show how the GIS is able to aid in defining and analyzing scenarios. These scenarios only represented a small portion of the capabilities of HTFS and Arc/Info. Arc/Info does has the ability to perform more stringent analysis, or the results could be further analyzed by exporting them from Arc/Info and using them in other models if the user desired. For example, the results of Scenario 2 (building a new Interstate) could be exported and cost/benefit analysis could be performed, or the internal rate of return of adding the link could be calculated. In the following chapter, the procedures used to accomplish this study will be summarized, and some directions for further research will be suggested.

		PTM	VMT
Single Unit 2 Axles (SU2)	Total Shift:	0	132.36
	Percent Shift:	0	1.08
	Forecasted:	0	12396.5
	Basecase:	0	12264.14
Single Unit 3 or 4 Axles (SU3&4)	Total Shift:	0	27.03
	Percent Shift:	0	1.06
	Forecasted:	0	2574.59
	Basecase:	0	2547.56
Conventional Semi 3 Axle (CS3)	Total Shift:	0	24.95
	Percent Shift:	0	1.13
	Forecasted:	0	2229.36
	Basecase:	0	2204.41
Conventional Semi 4 Axles (CS4)	Total Shift:	-591.07	-88.51
	Percent Shift:	-1.51	-1.5
	Forecasted:	38531.77	5798.16
	Basecase:	39122.84	5886.67
Conventional Semi 5 Axles (CS5)	Total Shift:	-1822.36	-153.68
	Percent Shift:	-0.27	-0.26
	Forecasted:	685513.9	58189.39
	Basecase:	687336.3	58343.07
Conventional Semi 6 Axles (CS6)	Total Shift:	-300.16	-24.68
	Percent Shift:	-1.39	-1.4
	Forecasted:	21349.1	1741.64
	Basecase:	21649.25	1766.32
Combined Tractor 4 Axles (CT4)	Total Shift:	-160.77	-18.82
	Percent Shift:	-3.65	-3.64
	Forecasted:	4245.9	497.88
	Basecase:	4406.68	516.7
Combined Tractor 5 Axles (CT5)	Total Shift:	-57.38	-3.98
	Percent Shift:	-1.36	-1.35
	Forecasted:	4176.86	291.12
	Basecase:	4234.24	295.1
Combined Tractor 6 Axles (CT6)	Total Shift:	-18.27	-1.19
	Percent Shift:	-1.27	-1.26
	Forecasted:	1422.48	92.91
	Basecase:	1440.75	94.09

LONG COMBINATION VEHICLES

Rocky Mountain Doubles (DS5)	Total Shift:	-2897.51	-202.16
	Percent Shift:	-8.4	-8.41
	Forecasted:	31588.58	2201.65
	Basecase:	34486.1	2403.8
Western Doubles (DS7)	Total Shift:	-33.87	-2.4
	Percent Shift:	-2.29	-2.29
	Forecasted:	1444.27	102.37
	Basecase:	1478.14	104.77
Turnpike Doubles (DS9)	Total Shift:	-45.69	-1.36
	Percent Shift:	-4.07	-4.07
	Forecasted:	1076.11	32.2
	Basecase:	1121.79	33.57
Triples (TRP)	Total Shift:	-14.7	-0.79
	Percent Shift:	-3.98	-4
	Forecasted:	354.64	18.84
	Basecase:	369.35	19.63
User Defined Alternatives (Undefined Triple) (XXX)	Total Shift:	-13.51	-0.72
	Percent Shift:	-3.94	-3.95
	Forecasted:	329.52	17.49
	Basecase:	343.03	18.21

Figure 5-12: Ten Percent Increase in Fuel Tax
National VMT and PTM Totals

CHAPTER 6

CONCLUSION AND DIRECTIONS FOR FURTHER RESEARCH

Geographic Information Systems have proven to be an outstanding platform for transportation modeling systems. GISs' database managers and data structures are optimized for spatial data, and much transportation data is spatial. GISs also have a large number of spatial analysis routines for managing and manipulating large databases, such as those used for transportation analysis. In addition, GISs' have excellent graphical display routines for viewing, editing, and analyzing data. Arc/Info is currently one of the most popular GISs because it has a comprehensive and diverse set of spatial analysis routines, is capable of managing a large number of different data types, and is easily customized via AML. The power and flexibility Arc/Info offers increases the efficiency and functionality of the Highway Traffic Forecasting System, which requires the management of large amounts of line, areal, and aspatial data.

Because specialized transportation models are continually being developed and enhanced, it is becoming common for these routines to be integrated within GISs, which typically don't contain them. HTFS is an example of one such transportation modeling system. This thesis has shown one approach to port and integrate transportation models within a GIS. In order to accomplish the integration of Arc/Info and HTFS, several steps were required and are outlined in the following.

Procedures for Integration

1) An understanding of HTFS's original usage and design within TransCAD was developed.

2) The data structures of TransCAD and Arc/Info were studied, and translation routines were written so HTFS's data could be imported into Arc/Info.

3) HTFS's routines were complied on a Unix workstation.

4) The data flow and data formats of HTFS's routines were studied, and AML routines were written to control the data flow and reproduce the required data formats.

5) HTFS's routines which output data formats unsuitable for import into Arc/Info were modified.

6) A comprehensive, easy-to-use GUI was designed and written in AML to aid the user in scenario definition, running the models, and importing the results.

7) Scenarios were developed to test the integration of HTFS and Arc/Info.

Of the above procedures, steps 4 and 6 can be considered the paramount steps in the integration phase. The fourth step, which involved producing the data formats and controlling the data flow, was the basis for being able to successfully run HTFS under Arc/Info. HTFS relies on data formats originally produced by TransCAD, as well as data formats written by HTFS routines which accessed data directly from TransCAD's data tables and then wrote them to a file. These data formats, and the routines which accessed data directly from TransCAD's data tables, were reproduced by using AML. The data formats which Arc/Info could not directly produce were created by AML extraction routines. Some of these extraction routines dump temporary null items from the data tables to create a space separation between items in the ASCII file. This was the original data format written by HTFS's extraction routines. Also, the different representation of

null numeric items by TransCAD and Arc/Info was addressed. In TransCAD, null items are represented as two dashed lines, which is not valid in Arc/Info. When a HTFS routine encounters two dashed lines, the value is defaulted to a -1, indicating an unknown value. Thus, it was necessary to default all null values to a -1 after the data was imported into Arc/Info.

Procedure 6, which required the development of the GUI, was extremely important. The GUI was designed to mimic the usage of HTFS within TransCAD so past users of the system would be able to easily adapt to HTFS running under Arc/Info. Additionally, the GUI provides all the interaction between the user and HTFS, so it needed to be intuitive. For this reason, the GUI was designed to lead the user through the process of creating and running a new scenario, or retrieving and running an existing scenario. The GUI leads the user through the steps required for defining a new scenario by displaying all data table categories separately (e.g., all state data tables are displayed as one group, and all network data tables are displayed as another group), and allows the user to select the appropriate data tables with a mouse. The only information a user is required to type in is the name of the scenario, a scenario description, and the name and description of a network if it is being extracted. From this information, the AML routines extract and format the data tables if necessary, write the scenario file, write the name of the scenario to another file, and invoke HTFS, which reads these files and processes the data. Once the models are run, the user is able to import the results to an Info file by selecting the import option from the GUI.

After the integration of Arc/Info and HTFS, several scenarios were developed and run to test the design and operation of the system. These scenarios, which were presented in Chapter 5, show some of the key features of HTFS's operation, as well as some different types of analysis within Arc/Info. Although HTFS has all its intended functionality within Arc/Info, several extensions could be made to HTFS which would make it a more comprehensive system. The directions for further research are discussed in the following section.

Directions for Further Research

As HTFS now operates under Arc/Info, it requires a fair amount of knowledge by the user of Arc/Info. For example, the user must know the proper procedures and commands for copying data tables, updating and editing data tables, and displaying results. Although TED and NETSHELL aid the user in many of these areas by allowing commands to be accessed through a menu (a major reason for designing HTFS to be used from them), the user must still have a strong understanding of the GIS. HTFS could be enhanced by further expanding the user interface to create a turn-key system where all the operations required by HTFS could be accessed from a comprehensive GUI. The GUI could include commands which copy data tables, build topology, update data table records, perform topological editing, format result files, and thematically display results. This could be done by developing a GUI which is started from the Arc module, manages all Arc level commands, starts NETSHELL, TED, and other modules, and calls additional

module specific menus when required. By further expanding the system, it could become a more powerful decision support system and be easily used by decision makers.

In addition to expanding the GUI, some of HTFS's routines could be modified to increase its analysis and modeling abilities. First, HTFS's report routines could be modified to report flows by commodity type. Currently, the report routines aggregate the commodity totals and then writes the results to a file. These routines could be modified to report the results before the aggregation occurs. Reporting the totals by commodity type would be useful in trying to determine if some commodities are more sensitive to policy changes than others, or if some commodities have a propensity to be shipped by certain truck classes. In addition to the report routines being modified, the netbuild routine should be modified to allow a much larger network, such as the NHPN Version 2.0, to be used. As discussed in Chapter 4, the NHPN Version 2.0, which consists of over 100,000 links, was modified to be used within HTFS. However, the netbuild utility cannot allocate enough memory to process the network. One problem here is that the netbuild utility creates a bi-directional network structure (i.e., each link is represented twice in opposite directions), thus doubling the size of the network. Therefore, the memory allocation routine should be evaluated and modified. By being able to process a larger network, HTFS could evaluate scenarios at a much finer level of detail.

A final modification could be made to NETSHELL's flow map utility. This utility was used to create the flow maps presented in Chapter 5, and its legend is scaled to the minimum and maximum values of the results field in the network data table. Since these values were changing after the models were run, the legends were scaled differently

on the basecase and projected maps, which was a bit deceiving. The legend should be modified so the user can designate the dividing point values on the legend.

Conclusion

This thesis has shown one approach to the integration of transportation models and Geographic Information Systems. Since the use of GIS is expanding at a rapid pace, it is likely that the integration of specialized models and GISs will continue. For this reason, the role of GIS-T professionals as integrators of these technologies is likely to increase in the future. Thus, in order to remain competent in the field, GIS-T professionals should have a thorough understanding and background in a number of different GISs, their data structures, and their capabilities. They also need a strong understanding of the models they are integrating. With this information, and the continued development of procedures for integrating these technologies, such as those demonstrated in this thesis, the GIS-T profession should continue to grow.

BIBLIOGRAPHY

BIBLIOGRAPHY

Burrough, P. (1986) Principles of Geographical Information Systems for Land Resources Assessment, New York, NY: Oxford University Press.

Caliper Corporation (1990) TransCAD Users Manual 1172 Beacon Street, Newton, MA 02161.

Chen, C. and C. Tseng (1994) "Integration of Transportation Models and The Geographic Information System in Transportation" Proceedings GIS-T '94, pp. 287-305.

Chin, S. et al. (1992) Highway Traffic Forecasting System 1992 Functional Description, (Draft) Prepared for FHWA Office of Policy Development by The Center for Transportation Analysis, Oak Ridge National Laboratory.

Densham, P. J. (1991) "Spatial Decision Support Systems" Geographic Information Systems, eds. David Maguire, Michael Goodchild, and David Rhind, New York, NY: John Wiley and Sons, Inc. pp. 403-412.

Dykstra, E. (1959) "A Note on Two Problems in Connection with Graphs", Numerische Mathematik, 1, pp. 269-271.

ESRI (1994) Arc/Info Data Management, Environmental Systems Research Institute, Inc. 380 New York Street, Redlands, CA. 92373.

ESRI (1993) ARC Macro Language: Developing Arc/Info Menus and Macros with AML, Environmental Systems Research Institute, Inc. 380 New York Street, Redlands, CA. 92373.

ESRI (1991) Understanding GIS: The Arc/Info Method, Environmental Systems Research Institute, Inc. 380 New York Street, Redlands, CA. 92373.

Gatrell, A. C. (1991) "Concepts of Space and Geographical Data", Geographic Information Systems, eds. David Maguire, Michael Goodchild, and David Rhind, New York, NY: John Wiley and Sons, Inc. pp. 119-134.

Hanson, S. (1986) The Geography of Urban Transportation, New York, NY: The Guilford Press. p. 61.

Haynes, K. and A. Fotheringham (1984) Gravity and Spatial Interaction Models, Beverly Hills, CA: Sage Publications.

Healey, R. G. (1991) "Database Management Systems" Geographic Information Systems, eds. David Maguire, Michael Goodchild, and David Rhind, New York, NY: John Wiley and Sons, Inc. pp. 251-267.

Henco Software (1991) INFO Reference Manual, Henco Software, Inc. Reprinted with permission by Environmental Systems Research Institute, Inc. 380 New York Street, Redlands, CA. 92373.

HNDES (1994) Highway Network Design and Evaluation System (HNDES) Version 1.0b: User Manual, The Transportation Center, University of Tennessee, Knoxville and The Center for Transportation Analysis, Oak Ridge National Laboratory.

Hogan, R. L. (1993) "Using the Spatial Data Transfer Standard with Transportation Data Bases", Proceedings GIS-T '93, pp. 42-47.

Hyman, G. (1969) "The Calibration of Trip Distribution Models", Environment & Planning, 1, pp. 105-112.

Lewis, S. M. (1995) Personal communication. 7 June 1995.

Lewis, S. (1990) "Comprehensive Transportation models and The Use of Geographic Information Systems", ITE Journal, March 1990, pp. 34-38.

Lewis, S. and S. McNeil (1986) "Developments in Microcomputer Network Analysis Tools for Transportation Planning", ITE Journal, October, 1986.

Liu, C. and B. A. Ralston (1990) The Bangladesh Transportation Modeling System, The International Bank for Reconstruction and Development, Washington, D.C.

Maguire, D. J. (1991) "An Overview and Definition of GIS", Geographic Information Systems, eds. David Maguire, Michael Goodchild, and David Rhind, New York, NY: John Wiley and Sons, Inc. pp. 9-20.

Middendorf, D. P. (1995) "Impact of Longer and Heavier Trucks on Shipper's Logistics Costs", CTA Intersections, 3, 1, March, 1995.

Neihaus, M. (1986) The Highway Traffic Forecasting System Users Guide, Prepared for FHWA, HPP-12. January, 1986.

NHPN (1994) The National Highway Planning Network (Version 2.0): Description, Schema, and File Structures, Prepared for FHWA, Office of Environment and Planning HEP-11, by The Transportation Center, University of Tennessee, Knoxville.

Okunieff, P. et al. (1994) "Location Referencing System for IVHS: A Progress Report", Proceedings GIS-T '94, pp. 72-78.

Openshaw, S. (1991) Developing Appropriate Spatial Analysis Methods for GIS", Geographic Information Systems, eds. David Maguire, Michael Goodchild, and David Rhind, New York, NY: John Wiley and Sons, Inc. pp. 389-402.

Oram, A. and S. Talbot (1991) Managing Projects with make, Sebastopol, CA: O'Reilly & Associates, Inc.

Peterson, B. (1984) Iterative Proportional Fitting Program Documentation and User's Guide - Version 1.0, (Draft Report) Energy Division, Oak Ridge National Laboratory.

Petzold, R. G. and D. M. Freund (1990) "Potential for Geographic Information Systems in Transportation Planning and Highway Infrastructure Management", Transportation Research Record, (1261): 1-1.

Raima Corporation (1990) db_Vista Reference Manual, 3245 146th Place S.E. Bellevue, WA 98007.

Ralston, B. A. (1994) "Object Oriented Spatial Analysis", P.Rogerson and S. Fotheringham, eds. in Spatial Analysis in GIS, London: Taylor and Francis. pp. 165-185

Ralston, B. A. et al. (1994) "A Spatial Decision Support System for Transportation Policy Analysis", Journal of Transport Geography, 2, 2, pp. 101-110

Ralston, B. A. et al. (1992) "Transportation Inland Logistics Manager", Software developed for the Agency for International Development, Famine Early Warning System.

Ralston, B. A. (1989) "Transportation Models, Geographic Databases, and Map Processing Models", Paper presented at the 85th Annual Meeting of the AAG, Baltimore, Md.

Ray, J. J. (1995) Personal communication. 19 February 1995.

Ray, J. J. (1992) The Highway Traffic Forecasting System V2.0B: Freight Network Policy Model Technical Description, (Draft Report) The Transportation Center, University of Tennessee, Knoxville.

Ray, J. J. (1992b) Freight Network Policy Model User's Manual, (Draft) Prepared for the FHWA, Transportation Studies Division by The Transportation Center, University of Tennessee and The Center for Transportation Analysis, Oak Ridge National Laboratory.

Ray, J. J. (1990) The Flow Resource Facility Problem, Unpublished Ph.D. Dissertation, Department of Geography, University of Tennessee, Knoxville.

Ray, J. J. and M. P. Armstrong. (1994) GIS-T: Progress and Prospects (Draft) Forthcoming paper, University of Iowa.

Ray, J. J. and F. Southworth (1992) The Highway Traffic Forecasting System V2.0B: Freight Network Model User's Manual, The Transportation Center, University of Tennessee, Knoxville.

Ray, J. J. and F. Southworth (1991) The Highway Traffic Forecasting System: DOS Version 1.0, The Transportation Center, University of Tennessee and The Center for Transportation Analysis, Oak Ridge National Laboratory.

REDS (1995) REDS: Route Editing System Version 1.0, Overview and Command Summary, The Transportation Center, University of Tennessee, Knoxville.

Reebie and Associates (1991) TRANSEARCH Database of U.S. Freight Flows, 1987, P.O. Box 1436, Greenwich, CT. 06836.

Simkowitz, H. J. (1990a) "Integrating Geographic Information System Technology and Transportation Models", Transportation Research Record (1271): 44-47.

Simkowitz, H. J. (1990b) "Using Geographic Information System Technology to Enhance the Pavement Management Process", Transportation Research Record (1261): 10-19

Southworth, F. (1978) "A Highly Disaggregated Modal Split Model - Some Tests", Environment & Planning, A, 10, pp. 795-812.

Southworth, F. and B. Peterson (1986) Review of the Highway Traffic Forecasting System, (Draft) Prepared for FHWA, Hpp-12. November, 1986.

TED (1995) Topological EDiting System (TED) Version 1.0: Overview and Command Summary, The Transportation Center, University of Tennessee, Knoxville.

TIUS (1987) Truck Inventory and Use Survey, United States Census Bureau, 1987.

Zhang, M. (1993) A Generator for Object Oriented Models of Spatial Analysis, unpublished Master's Thesis, Department of Geography, University of Tennessee, Knoxville.

Zhu, X. (1992) Integration of GIS and Transportation Analysis: An Interface between Arc/Info and FIELDS, unpublished Master's Thesis, Department of Geography, University of Tennessee, Knoxville.

APPENDICES

Appendix A-1: NHPN Link Items (From Ray, 1992)

DATA FIELD	TYPE	UNITS	DOMAIN	DESCRIPTION
ID	Long Int		Non-Negative	Designated BEA Region
Length	Float	Miles	Strictly Positive	Length of Link
Direction	Short Int		0 or 1	One-way or Two way link'
Anode	Long Int		Non-Negative	Origin Node of Link
Bnode	Long Int		Non-Negative	Destination of Node of Link
State	Short Int		1-56 with missing values	State FIPS Code
Speed	Float		Non-Negative or Missing Value	Average Speed on the Link (Not currently used)
Max. Op Wt.	Float	Pounds	Non-Negative or Missing Value	Maximum Operating Weight of a truck on the link.
Class	Short Int		1,2 or 3	Class designator: 1=Interstate, 2=Primary, 3=Other
Lanes	Short Int		Non-Negative or Missing Value	Number of lanes on link (Not currently used)
SubNet 1	Short Int		0, 1 or Missing Value	Permissions for SubNetwork 1
SubNet 2	Short Int		0, 1 or Missing Value	Permissions for SubNetwork 2
SubNet 3	Short Int		0, 1 or Missing Value	Permissions for SubNetwork 3
SubNet 4	Short Int		0, 1 or Missing Value	Permissions for SubNetwork 4
SubNet 5	Short Int		0, 1 or Missing Value	Permissions for SubNetwork 5
Bridge ID	Short Int		0, 1 or Missing Value	1=Use Bridge Formula to calculate truck operating weights
MAX SAW	Float	Pounds	0, 1 or Missing Value	Maximum weight for a Single Axle on the link
MAX TAW	Float	Pounds	Non-Negative or Missing Value	Maximum weight for a Tandem Axle on the link

Appendix A-2

NHPN Node Items

DATA FIELD	TYPE	UNITS	DOMAIN	DESCRIPTION
ID	Long Int		Non-Negative	Node Index
Longitude	Long Int	Decimal Degrees		
Latitude	Long Int	Decimal Degrees		
Name	Character			Name of Node from NHPN

BEA Items

DATA FIELD	TYPE	UNITS	DOMAIN	DESCRIPTION
ID	Long Int		1 to 181	Designated BEA Region
Area	Float	Square Mile	Strictly Positive	Area of BEA Region
BEA Name	Character			Name of BEA Region
Total Pop	Float			Total Population, 1986
TEMPMM	Float			Total Employment, Mid-March, 1986
TANPAY	Float			Total Annual Payments, 1986
TESTAB	Float			Total Establishments, 1986

Appendix A-3

State Items

DATA FIELD	TYPE	UNITS	DOMAIN	DESCRIPTION
ID	Long Int		1 to 56 with missing values	State FIPS Code
Area	Float	Square Miles	Strictly Positive	Area of State
State Name	Character			Name of State
GVW Limit	Float	Pounds	Non-Negative or Missing Value	Maximum Gross Weight of any truck operating within the state
SAW	Float	Pounds	Non-Negative or Missing Value	Maximum Single Axle Weight of any truck operating within the state
TAW	Float	Pounds	Non-Negative or Missing Value	Maximum Tandem Axle Weight of any truck operating within the state
SubNet 1	Short Int		0 or 1	Permissions on SubNetwork 1
SubNet 2	Short Int		0 or 1	Permissions on SubNetwork 2
SubNet 3	Short Int		0 or 1	Permissions on SubNetwork 3
SubNet 4	Short Int		0 or 1	Permissions on SubNetwork 4
Subnet 5	Short Int		0 or 1	Permissions on SubNetwork 5

Appendix A-4

Truck Items

DATA FIELD	TYPE	UNITS	DOMAIN	DESCRIPTION
ID	Long Int		1 to number of truck configuratio ns	index of ith truck configuration
Longitude	Long Int		Always Zero	Not Used
Latitude	Long Int		Always Zero	Not Used
Configurati on	Charact er			Name of the Truck Configuration
Cost Per Mile	Float	Dollars	Non-negative	Average cost per mile in dollars to operate the truck
Axle Group Length	Float	Feet	Non-negative	Average length from the front to the rear axle for the truck configuration
Axle Number	Float		Non-negative	Number of axles for the truck configuration
Max Op Gross Weight	Float	Pounds	Non-negative	Average maximum physical carrying capacity of the truck configuration
Logit Theta	Float			Parameter used in the submodal split model
Logit Delta	Float			Parameter used in the submodal split model
SubNet ID	Short Int		1 to 5	Index of the SubNetwork which the truck configuration is assigned to

Appendix A-5

Commodity Items

DATA FIELD	TYPE	UNITS	DOMAIN	DESCRIPTION
ID	Long Int		1 to number of commodities	Index of the ith commodity
Longitude	Long Int		Always Zero	Not Used
Latitude	Long Int		Always Zero	Not Used
Commodity Name	Character			Name of the commodity group
Model Type	Short Int		1=Origin-Constrained Model 2=Quasi-Constrained Model 3=Use observed data	Controls how commodity flows are calculated
Beta	Float		Non-negative	Distance Decay parameter for spatial interaction model
Gamma	Float			Modal Split parameter
Delta	Float			Modal Split parameter

Appendix B-1

TransCAD Conversion Routines

```
/*
** NAME:     T C L N 2 G E N . C
**
** PURPOSE:  Converts TransCAD link and/or node data into
**           ARC/INFO generate format.
**
** HISTORY:  Todd Raynor              Version 1.0
**           Geographic Information Systems Group
**           Pellissippi Research Institute
**           University of Tennessee, Knoxville, TN
**
**           PJ Nabors        1/19/94 Version 1.1
**           PJ Nabors        2/08/94 Version 1.2
**            - Convert coordinates to float!
**
**
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define HDRLEN  400
#define NAMELEN  80
#define RECLEN  256

FILE    *InFp;
FILE    *OutFp, *OutGeoFp;

void    FileConversion();
void    FixRecord();
int     ValidChar();

int     ArcType;

/**
*** MAIN()
**/
int   main(argc,argv)
 int   argc;
 char   *argv[];

{ /*Begin Main*/

 char  FtType[NAMELEN];
 char  TcFile[NAMELEN];
 char  AttFile[NAMELEN];
 char  GeoFile[NAMELEN];
```

```
/*** CHECK THE INPUT ***/

 if (argc < 5)
  {
  fprintf(stdout, "\n USAGE:  tcln2gen <ARC | NODE> <TransCADFileName> <AttributeFileName>
<GeoFileName>] \n");
  exit(1);
  }

 if (sscanf(argv[1], "%s", FtType) != 1)
  {
  fprintf(stdout, "\n USAGE:  tcmx2gen <ARC | NODE> <TransCADFileName>
<AttributeFileName> <GeoFileName>] \n");
  exit(1);
  }

 if ( (strcmp(FtType,"ARC") == 0) || (strcmp(FtType,"arc") == 0) )
  ArcType = 1;
 else ArcType = 0;

 if (sscanf(argv[2], "%s", TcFile) != 1)
  {
  fprintf(stderr, "\n USAGE:  tcln2gen <ARC | NODE> <TransCADFileName> <AttributeFileName>
<GeoFileName>] \n");
  exit(1);
  }

 if (sscanf(argv[3], "%s", AttFile) != 1)
  {
  fprintf(stderr, "\n USAGE:  tcln2gen <ARC | NODE> <TransCADFileName> <AttributeFileName>
<GeoFileName>] \n");
  exit(1);
  }

 if (sscanf(argv[4], "%s", GeoFile) != 1)
  {
  fprintf(stderr, "\n USAGE:  tcln2gen <ARC | NODE> <TransCADFileName> <AttributeFileName>
<GeoFileName>] \n");
   exit(1);
  }


 /*** OPEN THE FILES ***/

 if ((InFp = fopen(TcFile, "r")) == NULL)
  {
   perror(" Opening TransCAD Input File ");
   exit(1);
  }

 if ((OutFp = fopen(AttFile, "w")) == NULL)
  {
   perror(" Opening Attribute Output File ");
   exit(1);
  }
```

```
 if ((OutGeoFp = fopen(GeoFile, "w")) == NULL)
   {
    perror(" Opening Geography Output File ");
    exit(1);
   }

 fprintf(stdout, "\n[TCLN2GEN - Version 1.2 TRANSCAD Conversion Utility]\n");
 fprintf(stdout, " Geographic Information Systems Group\n");
 fprintf(stdout, " Pellissippi Research Institute, U.T. Knoxville\n\n");


 /*** CONVERT THE FILE ***/

 FileConversion();

 close(InFp);
 close(OutFp);
 close(OutGeoFp);

} /* End Main */


/*********************************************************
**                                                      **
** NAME:    F I L E C O N V E R S I O N ()              **
**                                                      **
** PURPOSE:  Converts a Link or Node file in            **
**                 TransCAD format to ARC/INFO import files.  **
**                                                      **
*********************************************************/
void      FileConversion()
{
 char  HBuffer[HDRLEN];
 char  Buffer[RECLEN];
 int       i, len;
 /* Define the input fields */
 long  Id, Geography;
 long     Xcoord, Ycoord;
 double DXcoord, DYcoord;


 i = 0;

 /* Read in Header */
 fgets(HBuffer, (HDRLEN - 1), InFp);

 /* Read first record */
 fgets(Buffer, (RECLEN - 1), InFp);

 /* Loop Until End of File */
 while (!feof(InFp))
   {
   FixRecord(Buffer);
```

```
    sscanf(Buffer,"%ld,%ld",&Id,&Geography);
    fprintf(OutFp,"%s\n",Buffer);

    if (ArcType)
     fprintf(OutGeoFp,"%ld\n",Id);
    else
     fprintf(OutGeoFp,"%ld ",Id);

    for (i=0; i < Geography;i++)
     {
      fgets(Buffer, (RECLEN - 1), InFp);
      sscanf(Buffer,"%ld,%ld", &Xcoord,&Ycoord);
      /* Change to float */
      DXcoord = (double) Xcoord / 1000000.0;
      DYcoord = (double) Ycoord / 1000000.0;
      fprintf(OutGeoFp,"%lf %lf\n", DXcoord,DYcoord);
     }
    if (ArcType)
     fprintf(OutGeoFp,"END\n");

    fgets(Buffer, (RECLEN - 1), InFp);
    } /*while*/

 fprintf(OutGeoFp,"END\n");

} /*FileConversion*/


/*********************************************************
**                                                      **
** NAME:    F I X R E C O R D ()                        **
**                                                      **
** PURPOSE:  Inputs a character string and reformats it **
**                  for ARC/INFO input.                 **
**                                                      **
*********************************************************/
void      FixRecord(NewStr)
 char *NewStr;
{
 int  i=0, j=0;
 int  Len=0;


 Len = strlen(NewStr);
 for (i = 0; i < Len; i++) {
  if (!ValidChar(NewStr[i])) {
          switch (NewStr[i]) {
          case '[': NewStr[i] = '(';
                    break;
          case ']': NewStr[i] = ')';
                    break;
          case '\"': NewStr[i] = '\'';
                    break;
          default:  NewStr[i] = ' ';
                    break;
```

```
        }/*switch*/

   }/*if !Val*/
   else if (i > 1) {
         if ((NewStr[i] == '-') && (NewStr[i-1] == '-')) {
          if (NewStr[i-2] == '\'') {
             NewStr[i-1] = ' ';
             NewStr[i] = ' ';
       }
      else if (NewStr[i-2] == ',') {
             NewStr[i-1] = '0';
             NewStr[i] = '0';
       }
      }
   }/*else*/
  }/*for*/

  NewStr[i] = '\0';


} /*FixRecord*/


/*********************************************************
**                                                      **
** NAME:    V A L I D C H A R ()                        **
**                                                      **
** PURPOSE:  Checks to see if the input character is a  **
**                valid NHPN Version 2.0 character.     **
**                Returns 1 if it is, 0 if it is not valid.  **
**                                                      **
*********************************************************/
int ValidChar(inchar)
 char inchar;
{

 if (isdigit(inchar))
  return(1);

 if (isalpha(inchar) && isupper(inchar))
  return(1);

 switch (inchar) {
     case '(':
     case ')':
     case '-':
     case '_':
     case ' ':
     case '\'':      /* For parsing purposes only */
     case ',':       /* For parsing purposes only */
     case '.':       /* For parsing purposes only */
           return(1);
           break;
     default:
           return(0);
```

```
            break;
        }

} /*ValidChar*/
```

Appendix B-2

```
/*
** NAME:    T C P Y 2 G E N . C
**
```

```
** PURPOSE:   Converts TransCAD  polygon features and data into
**                ARC/INFO generate format.
**
** HISTORY:   Todd Raynor                Version 1.0
**          Geographic Information Systems Group
**          Pellissippi Research Institute
**          University of Tennessee, Knoxville, TN
**
**                PJ Nabors          1/19/94 Version 1.1
**
**                Tino La Rosa    1/25/95 Version 1.2
**
** WARNIG:    Refer to DataConversion() for important information...
**
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define NAMELEN   80
#define RECLEN   256
#define EXTLEN    5

FILE    *InFp, *InlineFp, *IncentFp;
FILE    *OutFp, *OutlineFp, *OutlabelFp;

void    LineConversion();
void    LabelConversion();
void    DataConversion();


/**
*** MAIN()
**/
int    main(argc,argv,ch1,ch2,ch3)
 int    argc;
 char   *argv[];
 char   ch1[EXTLEN];
 char   ch2[EXTLEN];
 char   ch3[EXTLEN];


{ /*Begin Main*/

 char   TcGeoFile[NAMELEN];
 char   TcCntFile[NAMELEN];
 char   TcAttFile[NAMELEN];
 char   GeoFile[NAMELEN];


 /*** CHECK THE INPUT ***/

 if (argc < 5)
  {
```

```
  fprintf(stdout, "\n USAGE:  tcpy2gen <TCADGeoFileName> <TCADCentrdFileName>
<TCADAttrFileName> <OutCoverName] \n");
  exit(1);
  }

 if (sscanf(argv[1], "%s", TcGeoFile) != 1)
  {
  fprintf(stdout, "\n USAGE:  tcpy2gen <TCADGeoFileName> <TCADCentrdFileName>
<TCADAttrFileName> <OutCoverName] \n");
  exit(1);
  }

 if (sscanf(argv[2], "%s", TcCntFile) != 1)
  {
  fprintf(stdout, "\n USAGE:  tc2pygen <TCADGeoFileName> <TCADCentrdFileName>
<TCADAttrFileName> <OutCoverName] \n");
  exit(1);
  }

 if (sscanf(argv[3], "%s", TcAttFile) != 1)
  {
  fprintf(stdout, "\n USAGE:  tc2pygen <TCADGeoFileName> <TCADCentrdFileName>
<TCADAttrFileName> <OutCoverName] \n");
  exit(1);
  }

 if (sscanf(argv[4], "%s", GeoFile) != 1)
  {
  fprintf(stdout, "\n USAGE:  tcpy2gen <TCADGeoFileName> <TCADCentrdFileName>
<TCADAttrFileName> <OutCoverName] \n");
  exit(1);
  }

 /*** CREATE OUTPUT FILES ***/

 ch1=".arc";
 ch2=".lbl";
 ch3=".dat";

 /*  The following files will be created:

    OutlineFp=(GeoFile.ch1) as line file
    OutlabelFp=(GeoFile.ch2) as centroid file
    OutFp=(GeoFile.ch3) as attribute file
 */


 /*** OPEN THE FILES ***/


 if ((InlineFp = fopen(TcGeoFile, "r")) == NULL)
  {
   perror(" Opening TransCAD Line File ");
   exit(1);
  }
```

```
if ((IncentFp = fopen(TcCntFile, "r")) == NULL)
  {
   perror(" Opening TransCAD Centroid File ");
   exit(1);
  }

if ((InFp = fopen(TcAttFile, "r")) == NULL)
  {
   perror(" Opening TransCAD Attribute File ");
   exit(1);
  }

if ((OutlineFp = fopen(strcat(GeoFile,ch1), "w")) == NULL)
  {
   perror(" Opening Geography Output File ");
   exit(1);
  }

sscanf(argv[4], "%s", GeoFile);  /*  Reset the filename <GeoFile> */

if ((OutlabelFp = fopen(strcat(GeoFile,ch2), "w")) == NULL)
  {
   perror(" Opening Centroid Output File ");
   exit(1);
  }

sscanf(argv[4], "%s", GeoFile);  /*  Reset the filename <GeoFile> */

if ((OutFp = fopen(strcat(GeoFile,ch3), "w")) == NULL)
  {
   perror(" Opening Data Output File ");
   exit(1);
  }


fprintf(stdout, "\n[TCPY2GEN - Version 1.1 TRANSCAD Polygon Conversion Utility]\n");
fprintf(stdout, " Geographic Information Systems Group\n");
fprintf(stdout, " Pellissippi Research Institute, U.T. Knoxville\n\n");


printf("About to convert\n");


/*** CONVERT THE FILE */

LineConversion();
LabelConversion();
DataConversion();

close(InlineFp);
close(IncentFp);
close(InFp);
close(OutlineFp);
close(OutlabelFp);
```

```
 close(OutFp);

} /* End Main */


/*********************************************************
**                                                      **
** NAME:        L I N E C O N V E R S I O N ()          **
**                                                      **
** PURPOSE:   Converts a Link file in TransCAD           **
**                format to ARC/INFO import files.      **
**                                                      **
** Note:      To be associated to LabelConversion()      **
**                          for polygons conversion     **
**                                                      **
*********************************************************/

void      LineConversion()

{
 int      i;

 /* Define the input fields */

 int npoints, line_no;
 long idpoly1, idpoly2;
 double xcoord, ycoord;
 line_no = 1;
/* Read each set of coordinates  */

 while (fscanf(InlineFp, "%ld,%ld,%d",&idpoly1,&idpoly2,&npoints) != EOF)
 {
  fprintf(OutlineFp,"%d\n",line_no);

 /* write pair of coordinates until npoint */

  for (i=0;i<npoints;i++)
        {
         fscanf(InlineFp,",%lf,%lf", &xcoord,&ycoord);
         fprintf(OutlineFp,"%lf,%lf\n", xcoord,ycoord);
        }
  fprintf(OutlineFp,"END\n");
  line_no++;
 } /* end of loop */

fprintf(OutlineFp,"END\n");

} /* LineConversion() */


/*********************************************************
**                                                      **
** NAME:        L A B E L C O N V E R S I O N ()        **
**                                                      **
```

```
** PURPOSE:     Converts a Mexico centroid file in                **
**              TransCAD format to ARC/INFO import files.         **
** Note:        To be associated to LineConversion()              **
**                      for polygons conversion                   **
**                                                                **
*****************************************************************/

void  LabelConversion()

{
 int      i,j;
 char  c;
 /* Define the input fields */
 int id;
 long xcoord, ycoord;
 double x_coord, y_coord;
 i = 0;

 /* Read first data of the record in loop until EOF */

   while ( fscanf(IncentFp,"%d,",&id) != EOF)
    {

 /* Loop to skip the name of the region */

    j = 0; /* int counting the quotations */

    while (j < 2)
        {  /* read char until j = 2 */
        c = getc(IncentFp);
       if ( c == '\"')
          j++;
        }
    /* Read the rest of the record */

    fscanf(IncentFp,",%ld,%ld",&xcoord,&ycoord);

    /* coordinate conversion */

    x_coord = (float) xcoord / 1000000.0;
    y_coord = (float) ycoord / 1000000.0;
    fprintf(OutlabelFp,"%d,%lf,%lf\n",id,x_coord,y_coord);

    } /* end of the loop */

 fprintf(OutlabelFp,"END\n");

} /*LabelConversion*/


/*************************************************************
**                                                          **
** NAME:        D A T A C O N V E R S I O N ()               **
**                                                          **
```

```
** PURPOSE:   Converts an attribute file in TransCAD                **
**                format to ARC/INFO import files.                   **
**                                                                   **
** Note:      To be associated to LineConversion()                   **
**             and LabelConversion for polygons conversion           **
**                                                                   **
** WARNING: This subroutine has been disactivated because           **
**          the original data file is compatible with               **
**          Arc/Info format. The procedure just makes a             **
**          duplicate of the *.d0 data file called *.dat            **
**           for the correct execution of *.aml file                **
**           << MAKE ACTIVE IF *.d0 HAVE DIFFERENT FORMAT>>  **
**                                                                   **
***********************************************************/

void  DataConversion()

/*  Method of copying *.d0 to *.dat through data conversion */

{
  char c;
  while ((c = getc(InFp)) != EOF)
      {
       fprintf(OutFp, "%c",c);
      }

/* ACTIVATE THIS SECTION AND DISACTIVATE THE ABOVE ONE
      IN CASE OF UNCOMPATIBLE DATA FORMAT


{

 char  prevch = ' ';
 char  c;


 while ((c = getc(InFp)) != EOF)
      {
       if (prevch == ',' && c == '\n')
          fprintf(OutFp,"0");

       if (c == '\n')
          prevch =  ' ';

       if (prevch == ',' && c == ',')
          fprintf(OutFp,"0");

       if (c == ',')
          putc(' ',OutFp);

        else
          {
          if ( c != '\"')
           putc(c,OutFp);
           }
```

```
        prevch = c;
      }
*/

} /* DataConversion() */
```

Appendix C-1

HTFS's AML Routines

```
/*************************************************************/
/**
/**              H T F S . A M L
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:    Launches HTFS routines
/**
/** DATE:       8 March 1995
/**
/*************************************************************/
&args module
&severity &error &routine BAIL
&thread &focus &off &others

/*
/* Get the old environment...
/*
&s oldm [SHOW &messages]
&s oldamlpath [SHOW &amlpath]
&s oldmenupath [SHOW &menupath]
&s oldworkspace [SHOW &workspace]

/*
/* Now the new...
/*
&amlpath $HTFSDIR/aml
&menupath $HTFSDIR/aml

&s cancel .FALSE.
&s submod

&call MAIN
&call EXIT
&return

/*
/*
&ROUTINE MAIN
/*
/*

/* Display the main menu...

&thread &create htfs$htfsmenu &m htfs &pos &cc &stripe 'HTFS' ~
     &pinaction '&thread &focus &on &others'
 &s thread htfs$htfsmenu
 &call FOCUS_THREAD
```

```
&if %cancel% &then
  &call EXIT

/* What do they want to do...

&select %module%

 &when ABOUT
   &do
    &sys display htfs.gif
    &s module #
    &call MAIN
   &end

 &when FNET
   &do
    &if [SHOW PROGRAM] NE ARC &then
       &sys arc &r fnet
    &else
       &r fnet
    &s module #
    &call MAIN
   &end

 &when OGWR
   &do
    &if [SHOW PROGRAM] NE ARC &then
       &sys arc &r shell OPWT
    &else
       &r shell OPWT
    &s module #
    &call MAIN
   &end

 &when LLR
   &do
    &if [SHOW PROGRAM] NE ARC &then
       &sys arc &r shell LINK
    &else
       &r shell LINK
    &s module #
    &call MAIN
   &end

 &when IND
   &do

    &thread &create htfs$indi_modls &m indi_modls &pos &cc &stripe ~
    'Sub Models' &pinaction '&thread &focus &on &others'

    &s thread htfs$indi_modls
    &call FOCUS_THREAD

    &if %cancel% &then
       &do
```

```
      &s module #
      &s cancel .FALSE.
      &call MAIN
     &end

   &else
    &do
    &if NOT [NULL %submod%] AND [TRANSLATE %submod%] NE IMPRT &then
      &do
       &if [SHOW PROGRAM] NE ARC &then
         &do
          &sys arc &r shell %submod%
          &if %submod% = vmt &then
            &sys arc &r shell post
         &end
       &else
         &do
          &r shell %submod%
          &if %submod% = vmt &then
            &r shell post
         &end
      &end

    &if NOT [NULL %submod%] AND [TRANSLATE %submod%] = IMPRT &then
      &do
       &thread &create htfs$import &m import &pos &cc &stripe ~
       'Import File' &pinaction '&thread &focus &on &others'

       &s thread htfs$import
       &call FOCUS_THREAD

       &if NOT %cancel% &then
         &do
          &if [SHOW PROGRAM] NE ARC &then
            &sys arc &r import %file%
          &else
            &r import %file%
         &end
       &else
         &s cancel .FALSE.
      &end

    &s module #
    &call MAIN
   &end
  &end

 &when NET
   &do
    &thread &create htfs$network &m network &pos &cc &stripe ~
    'HTFS Networks' &pinaction '&thread &focus &on &others'

    &s thread htfs$network
    &call FOCUS_THREAD
```

```
    &if %cancel% &then
      &do
       &s module #
       &s cancel .FALSE.
       &call MAIN
      &end

     &if [EXISTS $HTFSDIR/%net_name% -file] &then
       &do
        &messages &popup
        &s res [QUERY 'Network file '%net_name' already exists. Overwrite' .FALSE.]
        &messages &on
        &if NOT %res% &then
          &do
           &s module #
           &call MAIN
          &end
       &end

      &if [SHOW PROGRAM] NE ARC &then
        &sys arc &r network %net_name% [QUOTE %net_descrpt%]
      &else
        &r network %net_name% [QUOTE %net_descrpt%]

    &s module #
    &call MAIN
   &end

 &otherwise /* Exit if invalid selection...
   &return

&end /* &select

&return

/*
/*
&ROUTINE FOCUS_THREAD
/*
/*
&thread &focus &off &self
&thread &focus &on &others

&if [SHOW &thread &exists %thread%] &then
  &thread &delete %thread%

&return

/*
/*
&ROUTINE EXIT
/*
/*
/* Return the environment...
/*
```

```
&messages %oldm%
&amlpath %oldamlpath%
&menupath %oldmenupath%
&workspace %oldworkspace%
&thread &focus &on &others
&return

/*
/*
&ROUTINE BAIL
/*
/*
&severity &error &ignore
&call EXIT
&return &inform An error occured. Exiting...
```

Appendix C-2

```
/************************************************************/
/**
/**              F N E T . A M L
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:    Controls the FNET Modeling routines ...
/**
/** DATE:       8 March 1995
/**
/************************************************************/
&severity &error &routines BAIL
&term 9999


/*
/* Existing or new scenario ...
/*
&messages &popup
 &s res [QUERY 'Create New Scenario']
&messages &on

&if %res% &then
   &do
    &s sno_type NEW
    &call NEW_SCENARIO
    &call WRITE_SCENARIO
    &call RUN_FNET
   &end

&else
   &do
    &s sno_type EXIST
    &call EXISTING_SCENARIO
    &call RUN_FNET
   &end

&return

/*
/*
&ROUTINE NEW_SCENARIO
/*
/*

/*
/* First get the scenario info...
/*
&thread &create htfs$newscenario &m new_sno &pos &cc &stripe ~
      'Enter the New Scenario Information' ~
       &pinaction '&thread &focus &on &others'
 &s thread htfs$newscenario
```

```
 &call FOCUS_THREAD

&if %cancel% &then
   &return &inform Exiting from FNET...

/*
/* Find out if they want to extract any covers...
/*
&messages &popup
&s res [QUERY 'Extract coverages' .FALSE.]
&messages &on

&if %res% &then
   &call EXTRACT_COVERS

/*
/* Now get the other spatial databases...
/*
&thread &create htfs$sta_trk_cdt &m sta_trk_cdt &pos &cc &stripe ~
      'Select the Truck, Commodity, and State Databases' ~
      &pinaction '&thread &focus &on &others'
  &s thread htfs$sta_trk_cdt
  &call FOCUS_THREAD

&if %cancel% &then
   &return &inform Exiting from FNET...

/*
/* Now get the OPWT and Economic Factors databases...
/*
&thread &create htfs$owt_econ &m opwt_econ &pos &cc &stripe ~
      'Select Operating Weight and Economic Factors' ~
      &pinaction '&thread &focus &on &others'
  &s thread htfs$owt_econ
  &call FOCUS_THREAD

&if %cancel% &then
   &return &inform Exiting from FNET...

/*
/* Now get the rest...
/*
&thread &create htfs$sply_dmnd_rail &m sply_dmnd_rail &pos &cc &stripe ~
      'Select Supply, Demand, and Rail' ~
      &pinaction '&thread &focus &on &others'
  &s thread htfs$sply_dmnd_rail
  &call FOCUS_THREAD

&if %cancel% &then
   &return &inform Exiting from FNET...

&s values new_snoname sno_descrpt net_name wght_fact state truck commodity ~
       owpt demand econ rail supply

&call CHECK_4_VALUES
```

```
&return

/*
/*
&ROUTINE EXISTING_SCENARIO
/*
/*
&thread &create htfs$exscenario &m exist_sno &pos &cc &stripe ~
      'Select an Existing Scenario' ~
       &pinaction '&thread &focus &on &others'
 &s thread htfs$exscenario
 &call FOCUS_THREAD

 &if %cancel% &then
    &return &inform Exiting from FNET...

&messages &popup
 &s res [QUERY 'Extract coverages']
&messages &on

&if %res% &then
    &call EXTRACT_COVERS

&return

/*
/*
&ROUTINE FOCUS_THREAD
/*
/*
&thread &focus &off &self
&thread &focus &on &others

&if [SHOW &thread &exists %thread%] &then
   &thread &delete %thread%


/*
/*
&ROUTINE EXTRACT_COVERS
/*
/*

&thread &create htfs$extract &m extract &pos &cc &stripe ~
      'Select the Databases to Extract' ~
       &pinaction '&thread &focus &on &others'
 &s thread htfs$extract
 &call FOCUS_THREAD

 &if %cancel% &then
    &return &inform Exiting from FNET...

/*
/* Run the extraction AML...
```

```
/*
 &if %sno_type% = NEW &then
   &do
    &if [SHOW PROGRAM] NE ARC &then
      &sys arc &r extract %new_snoname% %commodity% %truck% %states%
    &else
      &r extract %new_snoname% %commodity% %truck% %states%
   &end

 &if %sno_type% = EXIST &then
   &do
    &if [SHOW PROGRAM] NE ARC &then
      &sys arc &r extract [BEFORE %scenario% .sno] %commodity% %truck% %states%
    &else
      &r extract [BEFORE %scenario% .sno] %commodity% %truck% %states%
   &end

&return

/*
/*
&ROUTINE WRITE_SCENARIO
/*
/*

&if [EXISTS $HTFSDIR/srf/%new_snoname%.sno -file] &then
  &do
   &messages &popup
   &s res [QUERY 'WARNING: Scenario file '%new_snoname%'.sno already exists. Overwrite']
   &messages &on

   &if NOT %res% &then
     &return &inform Exiting from FNET...
  &end

&s fileunit [OPEN $HTFSDIR/srf/%new_snoname%.sno openstat -write]
  &if %openstat% ne 0 &then
    &do
     &type Error opening %new_snoname%.sno to write.
     &type Check if you have write permission to the htfs/srf directory.
     &return &inform Exiting from FNET...
    &end

&s wrtstat [WRITE %fileunit% [QUOTE %sno_descrpt%]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE [DATE -vfull]]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE %net_name%]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE %supply%]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE %demand%]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE %owpt%]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE %rail%]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE %econ%]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE [ENTRY %truck%]]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE [ENTRY %state%]]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE [ENTRY %commodity%]]]
&s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE %wght_fact%]]
```

```
 &s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE 181]]
 &s i 0
 &do &while %i% < 181
    &s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE %i%]]
    &s i %i% + 1
  &end

 &if %wrtstat% > 0 &then
    &do
     &type An error occured writing the scenario file.
     &return &inform Exiting from FNET...
    &end

 &s clse [CLOSE %fileunit%]
&return

/*
/*
&ROUTINE CHECK_4_VALUES
/*
/*

&do I &list %values%
   &if [NULL [VALUE [VALUE I]]] &then
      &return &inform An invalid selection was made. Exiting from FNET...
   &end
&return

/*
/*
&ROUTINE RUN_FNET
/*
/*

/*
/* Get the htfs directory from the enviornment variable
/*
&messages &off &all
 &s work [SHOW workspace]
 &workspace $HTFSDIR
 &s htfsdir [SHOW workspace]
 &workspace %work%
&messages &on

&if %sno_type% = NEW &then
   &s sno_file %htfsdir%/srf/%new_snoname%.sno

&if %sno_type% = EXIST &then
   &s sno_file %scenario%

&messages &popup
 &s res [QUERY 'Run FNET Now']
&messages &on

&if NOT %res% &then
```

```
   &return

/*
/* Write the scenario file name to config.tmp
/*
&if %res% &then
   &do
    &s fileunit [OPEN $HTFSDIR/tmp/config.tmp openstat -write]
    &if %openstat% ne 0 &then
       &do
        &type Error opening config.tmp to write.
        &type Check if you have write permission to the htfs/srf directory.
        &return &inform Exiting from FNET...
       &end

      &s wrtstat [WRITE %fileunit% [QUOTE %sno_file%]]
      &s clse   [CLOSE %fileunit%]

    &if [SHOW PROGRAM] NE ARC &then
       &sys arc &r shell MAIN
    &else
       &r shell MAIN

/*
/* Go back to the main menu...
/*
&if [SHOW PROGRAM] NE ARC &then
   &sys arc &r htfs
&else
   &r htfs

&return

/*
/*
&ROUTINE BAIL
/*
/*
&severity &error &ignore
&s clse [CLOSE -ALL]
&return &inform An error occured. Exiting from FNET...
```

Appendix C-3

```
/*************************************************************/
/**
/**              N E T W O R K . A M L
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:   Unloads the attributes needed for NETBUILD.C
/**
/** DATE:      12 March 1995
/**
/*************************************************************/
&args net_name net_descrpt
&severity &error &routine BAIL

&if [NULL %net_name%] | [NULL %net_descrpt%] &then
  &do
   &type Error: A variable was not passed to network.aml...
   &return
  &end

/*
/* Set the aml path...
/*
&s amlpath aml/dev

/* Go to the networks directory...
&messages &off &all
&type Unloading network...
&s work [SHOW workspace]
workspace $HTFSDIR/networks

/* Get rid of the temporary link file...
&if [EXISTS $HTFSDIR/transcad/links.tmp -file] &then
  &s del [DELETE $HTFSDIR/transcad/links.tmp -file]

/* Start tables and unload the data...
TABLES

 SEL [ENTRY %net_name%].aat

/* If the impedance item doesn't exist, then it's NHPNv1 so....
&if NOT [ITEMINFO %net_name% -ARC IMPEDE -EXISTS] &then
  &do
    UNLOAD $HTFSDIR/temp/links.out ~
     LINK_ID,ANODE,BNODE,LENGTH2,~
     STATE,SPEED,MAX_OPT_WT,CLASS,LANES,~
     SUBNET1,SUBNET2,SUBNET3,SUBNET4,~
     SUBNET5,BRIDGE_ID,MAX_TAW,MAX_SAW,TEMP ~
     DELIMITED
  &end

&else /* This is NHPN Version 2.0
```

```
  &do
    &messages &popup
    &s res [QUERY 'Extract NHS']
    &messages &off &all

    &if %res% &then
      &do
       &type Reselecting NHS...
       RESELECT NHS > 0
      &end

    UNLOAD $HTFSDIR/transcad/links.tmp ~
     RECID,FNODE,TNODE,MILES,~
     STFIPS,SPEED,MAX_OPT_WT,CLASS,LANES,~
     SUBNET1,SUBNET2,SUBNET3,SUBNET4,~
     SUBNET5,BRIDGE_ID,MAX_TAW,MAX_SAW,IMPEDE ~
     DELIMITED
  &end


 Q STOP

/* Change to the original directory...
workspace %work%
&messages &on

/* Now write the solution.tmp file...

&if [EXISTS $HTFSDIR/transcad/solution.tmp -file] &then
  &s del [DELETE $HTFSDIR/transcad/solution.tmp -file]

&s fileunit [OPEN $HTFSDIR/transcad/solution.tmp openstat -write]
  &if %openstat% ne 0 &then
    &do
     &type Error opening solution.tmp to write.
     &return &inform Exiting from network.aml...
    &end

  &s wrtstat [WRITE %fileunit% [QUOTE Description:]]
  &s wrtstat %wrtstat% + [WRITE %fileunit% %net_descrpt%]
  &s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE File name]]
  &s wrtstat %wrtstat% + [WRITE %fileunit% [QUOTE [ENTRY %net_name%]]]

  &if %wrtstat% > 0 &then
   &do
    &type An error occured writing the solution.tmp file.
    &return &inform Exiting from network.aml...
   &end

 &s clse [CLOSE %fileunit%]

/* Now run netbuild...

&if [SHOW PROGRAM] NE ARC &then
  &sys arc &r $HTFSDIR/%amlpath%/shell netbuild
```

```
&else
  &r $HTFSDIR/%amlpath%/shell netbuild

&return

/*
/*
&routine BAIL
/*
/*
&severity &error &ignore

&if [SHOW program] = TABLES &then
  Q STOP

&if [SHOW workspace] NE %work% &then
  workspace %work%
&messages &on

&return &inform An error occured dumping network.  Exiting...
```

Appendix C-4

```
/*****************************************************************/
/**
/**                EXTRACT.AML
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:   Extracts Commodity, Trucks, and States databases to ASCII...
/**
/** DATE:      13 March 1995
/**
/*****************************************************************/
&args xtrct_sno_name commodity truck states
&severity &error &routine BAIL

/* change to the polygon directory...
&messages &off &all
&s work [SHOW workspace]
workspace $HTFSDIR/polygon
TABLES

&if NOT  %commodity% = # &then
  &do
    &type Extracting commodities...
    SEL [AFTER [ENTRY %commodity%] arc!]
    UNLOAD $HTFSDIR/srf/[ENTRY %xtrct_sno_name%].cdt MODEL BETA GAMMA DELTA
SELECTED ~
    TMP NAME COLUMNAR $HTFSDIR/temp/format.fil INIT
  &end

&if NOT %truck% = # &then
  &do
    &type Extracting trucks...
    SEL [AFTER [ENTRY %truck%] arc!]
    UNLOAD $HTFSDIR/srf/[ENTRY %xtrct_sno_name%].trk CONFIGURATION
COST_PER_MILE ~
    AXLE_GROUP LGTH_AXLE_# MAX_OP_WT LOGIT_THETA LOGIT_DELTA ~
    SUB_NET_ID SELECTED COLUMNAR $HTFSDIR/temp/format.fil INIT
  &end

&if NOT %states% = # &then
  &do
    &type Extracting states...
    SEL [ENTRY %states%].pat
    RES STATEFIPS NE 0
    UNLOAD $HTFSDIR/srf/[ENTRY %xtrct_sno_name%].sta STATEFIPS TMP GVW_LMT TMP
~
    SAW TMP TAW ~
    SUBNET1 SUBNET2 SUBNET3 SUBNET4 SUBNET5 TMP NAME COLUMNAR ~
    $HTFSDIR/temp/format.fil INIT
  &end

Q STOP
```

```
workspace %work%
&messages &on

&return

/*
/*
&routine BAIL
/*
/*
&severity &error &ignore

&if [SHOW program] = TABLES &then
  Q STOP

&if [SHOW workspace] NE %work% &then
  workspace %work%
&messages &on

&return &inform An error occured extracting databases.  Exiting...
```

Appendix C-5

```
/*****************************************************************/
/**
/**              S H E L L . A M L
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:   Runs the desired HTFS routine.
/**          It is needed because the POST/IPF
/**          routine needs to be run from the
/**          progs directory.
/**
/** DATE:      12 March 1995
/**
/*****************************************************************/
&args routine
&severity &error &routine BAIL

&messages &off &all
 &s work [SHOW workspace]
 &workspace $HTFSDIR/progs
&messages &on

&sys shell %routine%

&messages &off &all
 &workspace %work%
&messages &on
&return

/*
/*
&ROUTINE BAIL
/*
/*
&severity &error &ignore

&if [SHOW workspace] NE %work% &then
  &do
   &messages &off &all
    &workspace %work%
   &messages &on
  &end
&return &inform An error occured changing to $HTFSDIR/progs directory...
```

Appendix C-6

```
/*************************************************************************/
/**
/** IMPORT.AML
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:    Reads result's file header and defines an INFO file
/**                 for the results to be imported into.
/**
/** DATE:       20 March 1995
/**
/*************************************************************************/
&args file
&severity &error &routine BAIL

&if [NULL %file%] &then
    &return &inform Error: File name was not passed.

&if NOT [EXISTS $HTFSDIR/txt/%file%.hdr] &then
  &return &inform Error: File %file%.hdr was not found.

/* Get the header...
&s fileunit [OPEN $HTFSDIR/txt/%file%.hdr openstat -read]

&if %openstat% ne 0 &then
  &return &inform Error: Could not open %file%.hdr.

&s header [READ %fileunit% readstat]

&if %readstat% ne 0 &then
  &return Error: Could not read %file%.hdr.

&s clse [CLOSE %fileunit%]

/* Change to the appropriate directory...
&messages &off &all
&s work [SHOW workspace]

&select %file%
  &when ptm, vmt
    &workspace $HTFSDIR/polygon

  &when linktrip, linktons
    &workspace $HTFSDIR/networks

  &otherwise
    &return &inform Error: Unknown file name %file%.txt
&end

/* Find out how many items are in the string...
&s lgth1 [LENGTH %header%]
&s header [SUBST %header% , ,,]
```

```
&s lgth2 [LENGTH %header%]
&s NumElemnts [CALC %lgth2% - %lgth1% + 1]
&s header [SUBST %header% ,, ,]

/* Get rid of the blanks and quotes...
&s header [subst [QUOTE %header%] ' ' '']
&s header [subst %header% " '']

/* Define the table...

TABLES
 &s i 1

 &if [EXISTS %file%.txt -info] &then
    KILL %file%.txt

 &type Defining %file%.txt...

 DEFINE %file%.txt
 &do &while %i% <= %NumElemnts%

    &select %file%
     &when linktrip, linktons
       &if %i% = 1 &then
          [EXTRACT %i% %header%],4,5,b
       &else
          [EXTRACT %i% %header%],8,12,f,2

     &when vmt, ptm
       &if %i% = 1 &then
          [EXTRACT %i% %header%],2,2,i
       &else
          [EXTRACT %i% %header%],8,12,f,4
    &end

  &s i %i% + 1
 &end

 [UNQUOTE ' ']
 &type Importing data...
 ADD FROM $HTFSDIR/txt/%file%.txt

Q STOP

&messages &off &all
&workspace %work%
&messages &on

&type Completed...
&return

/*
/*
&ROUTINE BAIL
/*
```

```
/*
&severity &error &ignore

&if [SHOW PROGRAM] = TABLES &then
  Q STOP

&messages &off &all
&workspace %work%
&messages &on
&s clse [CLOSE %fileunit%]
&return &inform An error occured creating INFO file...
```

Appendix C-7

```
/*************************************************************************/
/**
/**          T R U C K S H I F T . A M L
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:   Calculates the shift in truck classes and prints a report file.
/*
/** DATE:      25 May 1995
/**
/*************************************************************************/
&args forecast_file base_file stat_file dbtype
&severity &error &routine BAIL

&if [null %forecast_file%] | [null %base_file%] | [null %stat_file%] | [NULL %dbtype%] &then
  &return &inform \Usage: truckshift <results_file> <basecase_file> <summary_file> <LINK |
STATE>\

/*
/* Validate the type of database...
/*
&if [TRANSLATE %dbtype%] NE LINK &then
  &do
   &if [TRANSLATE %dbtype%] NE STATE &then
     &return &inform \Unknown database type %dbtype%...\
  &end

&if [TRANSLATE %dbtype%] = STATE &then
 &s trk_list SU2 SU3&4 CS3 CS4 CS5 CS6 CT4 CT5 CT6 DS5 DS7 DS9 TRP XXX
&else
 &s trk_list CS4 CS5 CS6 CT4 CT5 CT6 DS5 DS7 DS9 TRP XXX

&messages &off
&s prog [SHOW PROGRAM]

&if %prog% NE ARC AND %prog% NE TABLES &then
    &do
     &messages &on
     &return &inform Truck Shift is run from ARC or Tables...
    &end

&if %prog% = ARC &then
  TABLES

&if [EXISTS %stat_file% -INFO] &then
 &do
  &type
  &if [QUERY 'Statistics file '%stat_file%' already exists. Overwrite'] &then
    KILL %stat_file%
  &else
   &do
    &messages &on
```

```
    &return &inform Exiting...
  &end
 &end

&if NOT [NULL [SHOW SELECT]] &then
  SELECT

/*
/* Find out if they want stats on a sub-set...
/*
&s continue .TRUE.
&s DoSelect .FALSE.
&type
&if [QUERY 'Do you want to reselect any items'] &then
  &do &while %continue%
    &s reselect [RESPONSE 'Enter the SQL']
    &type \You entered: %reselect%\
    &if [QUERY 'Is this correct'] &then
      &do
       &s continue .FALSE.
       &s DoSelect .TRUE.
      &end
  &end

/*
/* Set up the statistics file...
/*
&s numfields 0
DEFINE %stat_file%
&do I &list %trk_list%
 %I%_shift 8 12 F 2
 %I%_shft_prcnt 8 12 F 2
 %I%_tot_new 8 12 F 2
 %I%_tot_base 8 12 F 2
 &s numfields %numfields% + 4
&end
[UNQUOTE ' ']

/*
/* Intialize the database...
/*
&s k 1
ADD
&do &while %k% <= %numfields%
  0
  &s k %k% + 1
&end
[UNQUOTE ' ']

/*
/* Get the basecase total for each truck type...
/*
&type \Calculating statistics for the basecase...\
&s slctd_file %base_file%
&s field base
```

```
&CALL GET_STATS

/*
/* Get the forecasted results for each truck type...
/*
&type \Calculating forecasted results...\
&s slctd_file %forecast_file%
&s field new
&CALL GET_STATS

/*
/* Calculate the shift...
/*
&type Calculating total shift...
&do I &list %trk_list%
  CALCULATE %I%_SHIFT = %I%_TOT_NEW - %I%_TOT_BASE
&end

/*
/* Calculate the percent shift...
/*
&type Calculating percent shift...

&do I &list %trk_list%
  /* Dividing by zero is bad so...
  RESELECT %I%_TOT_BASE > 0
  &if [show number select] > 0 &then
  CALCULATE %I%_SHFT_prcnt = 100 * %I%_SHIFT / %I%_TOT_BASE
  ASELECT
&end

/*
/* Display the results...
/*
&messages &on
&type \Here is the summary:\
&sys sleep 2
list

&type
&if [QUERY 'Write data to file'] &then
  &call UNLOAD_DATA

&if %prog% = ARC &then
  Q STOP

&return

/*
/*
&ROUTINE GET_STATS
/*
/*
&if [EXISTS xxstat.out -FILE] &then
  &s del [DELETE xxstat.out -FILE]
```

```
&do I &list %trk_list%
 &type ' Calculating '%I%'...'
 &if [EXISTS xxSTAT.TMP -INFO] &then
    KILL xxSTAT.TMP

 SELECT [VALUE slctd_file]

 /*
 /* If they chose to use a subset...
 /*
 &if %DoSelect% &then
  &do
    ASELECT
    &type Performing: %reselect%
    [UNQUOTE %reselect%]
    &if [SHOW NUMBER SELECT] = 0 &then
       &do
        &type Error is selection statement...
        &call BAIL
       &end
  &end

 STATISTICS # xxSTAT.TMP
  SUM %I%
 END

 /* Work around to no cursors in Tables...
 SELECT xxSTAT.TMP
 UNLOAD xxstat.out SUM-%I%

 &s unit [OPEN xxstat.out ok -read]
 &s sum [READ %unit% ok]
 &s ok [CLOSE %unit%]
 &s del [DELETE xxstat.out -file]

 SELECT %stat_file%
 CALC %I%_tot_%field% = [UNQUOTE %sum%]

&end
&return

/*
/*
&ROUTINE UNLOAD_DATA
/*
/*
&messages &off
&s out_file [LOCASE [BEFORE %stat_file% .]]

&if [EXISTS %out_file%.out -FILE] &then
   &do
    &if [QUERY 'File '%out_file%'.out already exists. Overwrite'] &then
       &s del [DELETE %out_file%.out -file]
    &else
```

```
      &return
   &end

/* Print the info to the file...
&type Writing data to %out_file%.out...
&watch xx%out_file%.tmp
list
&watch &off

/* Get rid of the junk...
&sys sed s/'\!'/''/ xx%out_file%.tmp > xx%out_file%.tmp2
&sys sed s/'='/''/ xx%out_file%.tmp2 > %out_file%.out

/* Delete the temporary files...
&s del [DELETE xx%out_file%.tmp -FILE]
&s del [DELETE xx%out_file%.tmp2 -FILE]
&messages &on
&return

/*
/*
&ROUTINE BAIL
/*
/*
&severity &error &ignore
&if %prog% = ARC &then
   Q STOP
&messages &on
&return &inform An error occured. Exiting...
```

Appendix D-1

HTFS's AML Menus

```
7
/*************************************************************/
/**
/**                HTFS.MENU
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:   Launches HTFS routines
/**
/** DATE:      8 March 1995
/**
/*************************************************************/

 HIGHWAY TRAFFIC FORECASTING SYSTEM

 ================================================

   %ck1   About Highway Traffic Forecasting System

   %ck2   Freight Network Policy Model

   %ck3   Operating Gross Weight Report

   %ck4   Link Load Report

   %ck5   Individual Submodels

   %ck6   Network Utility

 ================================================

              %button
%ck1 CHECKBOX ck1 RETURN '&s module ABOUT; &s ck1 .FALSE.; ~
                &thread &focus &on &others'
%ck2 CHECKBOX ck2 RETURN '&s module FNET; &s ck2 .FALSE.; ~
                &thread &focus &on &others'
%ck3 CHECKBOX ck3 RETURN '&s module OGWR; &s ck3 .FALSE.; ~
                &thread &focus &on &others'
%ck4 CHECKBOX ck4 RETURN '&s module LLR; &s ck4 .FALSE.; ~
                &thread &focus &on &others'
%ck5 CHECKBOX ck5 RETURN '&s module IND; &s ck5 .FALSE.; ~
                &thread &focus &on &others'
%ck6 CHECKBOX ck6 RETURN '&s module NET; &s ck6 .FALSE.; ~
                &thread &focus &on &others'
%button BUTTON 'CANCEL' &s cancel .TRUE.; &thread &focus &on &others
%forminit &s ck1 .FALSE.; &s ck2 .FALSE.; &s ck3 .FALSE.; &s ck4 .FALSE.; ~
      &s ck5 .FALSE.; &s ck6 .FALSE.; ~
```

Appendix D-2

```
7
/*************************************************************/
/**
/**          I N D I _ M D L E S . M E N U
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:    Launches HTFS routines
/**
/** DATE:       8 March 1995
/**
/*************************************************************/

 H I G H W A Y  T R A F F I C  F O R E C A S T I N G  S Y S T E M

    ( I N D I V I D U A L  S U B - M O D E L S )

=================================================

   %ck1  Spatial Interaction Model

   %ck2  Multi-Objective Truck Routing Model

   %ck3  Modal (Rail/Truck) Split Model

   %ck4  Sub Modal (Truck Competion) Model

   %ck5  Vehicle Miles of Travel Forecast

   %ck6  Import Text Files

=================================================

              %button
%ck1 CHECKBOX ck1 RETURN '&s submod spin; &s ck1 .FALSE.; ~
              &thread &focus &on &others'
%ck2 CHECKBOX ck2 RETURN '&s submod path; &s ck2 .FALSE.; ~
              &thread &focus &on &others'
%ck3 CHECKBOX ck3 RETURN '&s submod modesplt; &s ck3 .FALSE.; ~
              &thread &focus &on &others'
%ck4 CHECKBOX ck4 RETURN '&s submod trcksplt; &s ck4 .FALSE.; ~
              &thread &focus &on &others'
%ck5 CHECKBOX ck5 RETURN '&s submod vmt; &s ck5 .FALSE.; ~
              &thread &focus &on &others'
%ck6 CHECKBOX ck6 RETURN '&s submod imprt; &s ck6 .FALSE.; ~
              &thread &focus &on &others'
%button BUTTON 'CANCEL' &s cancel .TRUE.; &thread &focus &on &others
%forminit &s ck1 .FALSE.; &s ck2 .FALSE.; &s ck3 .FALSE.; ~
      &s ck4 .FALSE.; &s ck5 .FALSE.; &s ck6 .FALSE.
```

Appendix D-3

```
7
/*************************************************************/
/**
/**          I M P O R T . M E N U
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:    Imports results files
/**
/** DATE:       20 March 1995
/**
/*************************************************************/

        Select the Files to Import

 ========================================

    %ck1   Link Trips

    %ck2   Link Tons

    %ck3   Vehicle Miles of Travel

    %ck4   Payload Ton Miles

 ========================================

           %button
%ck1 CHECKBOX ck1 RETURN '&s file linktrip; &s ck1 .FALSE.; ~
                  &thread &focus &on &others'
%ck2 CHECKBOX ck2 RETURN '&s file linktons; &s ck2 .FALSE.; ~
                  &thread &focus &on &others'
%ck3 CHECKBOX ck3 RETURN '&s file vmt; &s ck3 .FALSE.; ~
                  &thread &focus &on &others'
%ck4 CHECKBOX ck4 RETURN '&s file ptm; &s ck4 .FALSE.; ~
                  &thread &focus &on &others'
%button BUTTON 'CANCEL' &s cancel .TRUE.; &thread &focus &on &others
%forminit &s cancel .FALSE.; &s ck1 .FALSE.; &s ck2 .FALSE.; &s ck3 .FALSE.; ~
        &s ck4 .FALSE.
```

Appendix D-4

```
7
/*************************************************************/
/**          O P W T _ E C O N . M E N U
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:    Prompts the user to select OPWT and ECONOMIC Databases...
/**
/** DATE:       8 MARCH 1995
/**
/*************************************************************/

   ^Operating Weight:

    %disp1

   ^Economic Factors:

    %disp2

 ================================================

    ^Select the OPWT and Economic Factors Databases:

   ^Operating Weight:    ^Economic Factors:

   %input1              %input2




          %button1   %button2

%disp1 DISPLAY owpt 45
%disp2 DISPLAY econ 45
%input1 INPUT owpt 20 ~
    TYPEIN NO SCROLL YES ROWS 5 FILE '$HTFSDIR/srf/*.ogw' ~
    CHARACTER
%input2 INPUT econ 20 ~
    TYPEIN NO SCROLL YES ROWS 5 FILE '$HTFSDIR/srf/*.txt' ~
    CHARACTER
%button1 BUTTON 'ACCEPT' &s cancel .FALSE.; &thread &focus &on &other
%button2 BUTTON 'CANCEL' &s cancel .TRUE.;  &thread &focus &on &other
```

Appendix D-5

```
7
/*************************************************************/
/**
/**          E X I S T S C E N A R I O . M E N U
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:        Allows the user to select an existing scenario ...
/**
/** DATE:           8 MARCH 1995
/**
/*************************************************************/

 Scenario Name:

 %disp1

 Scenario Date:

 %disp2

 ====================================

     Select a Scenario:

     %input



       %button1  %button2

%disp1 DISPLAY sno_name 40
%disp2 DISPLAY sno_date 40
%input INPUT scenario 25 RETURN '&s stat [OPEN %scenario% openstat -read]; ~
    &s sno_name [UNQUOTE [READ %stat% readstat]]; ~
    &s sno_date [UNQUOTE [READ %stat% readstat]]; &s clse [CLOSE %stat%]' ~
    TYPEIN NO SCROLL YES ROWS 4 FILE '$HTFSDIR/srf/*.sno' ~
    CHARACTER
%button1 BUTTON 'ACCEPT' &s cancel .FALSE.; &thread &focus &on &other
%button2 BUTTON 'CANCEL' &s cancel .TRUE.; &thread &focus &on &other
```

Appendix D-6

```
7
/*************************************************************/
/**
/** N E W _ S N O . M E N U
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:    Sets up a new scenario file ...
/**
/** DATE:       9 March 1995
/**
/*************************************************************/

 ^Enter the Scenario File Name:

  %input1

 ^Enter the Scenario Description:

  %input2

     ^Select the Network:

     %input3



   ^Enter the Spatial Interaction Weighting Factor:

     %slider

        %button1  %button2

%input1 INPUT new_snoname 15 TYPEIN YES SCROLL NO SIZE 15 REQUIRED CHARACTER
%input2 INPUT sno_descrpt 40 TYPEIN YES SCROLL NO SIZE 40 REQUIRED CHARACTER
%input3 INPUT net_name    30 TYPEIN NO SCROLL YES ROWS 4 ~
     FILE '$HTFSDIR/srf/*.nwk' REQUIRED
%slider SLIDER wght_fact 30 INITIAL .5 STEP .1 REAL 0.0 1.0
%button1 BUTTON 'ACCEPT' &s cancel .FALSE.; &thread &focus &on &others
%button2 BUTTON 'CANCEL' &s cancel .TRUE.; &thread &focus &on &others
```

Appendix D-7

```
7
/*************************************************************/
/**
/**          N E T W O R K . M E N U
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:    Gets the name of the network to dump...
/**
/** DATE:       8 MARCH 1995
/**
/*************************************************************/

 Network Name:

 %disp1

 ^Enter a Network Description:

  %input1

 ===============================================

         ^Select the Network to Extract:

          %input2



            %button1  %button2

%disp1 DISPLAY net_name 50
%input1 INPUT net_descrpt 45 TYPEIN YES SCROLL NO SIZE 40 REQUIRED CHARACTER
%input2 INPUT net_name 25 ~
    TYPEIN NO SCROLL YES ROWS 4 REQUIRED COVER '$HTFSDIR/networks/*' -arc
%button1 BUTTON 'ACCEPT' &s cancel .FALSE.; &thread &focus &on &other
%button2 BUTTON 'CANCEL' &s cancel .TRUE.; &thread &focus &on &other
```

Appendix D-8

```
7
/*************************************************************/
/**
/**          E X T R A C T . M E N U
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:    Selects the databases to extract ...
/**
/** DATE:       8 MARCH 1995
/**
/*************************************************************/

    Trucks Database:

     %disp1

    Commodities Database:

     %disp2

    States Database:

     %disp3

    ==========================================================

                ^Select the Databases to Extract:

   ^Trucks:          ^Commodity:          ^States:

   %input1           %input2              %input3



      %1 Extract          %2 Extract          %3 Extract

                   %button1  %button2

%disp1 DISPLAY truck 65
%disp2 DISPLAY commodity 65
%disp3 DISPLAY states 65
%1 CHECKBOX ck1 RETURN '&if %ck1% &then; &s truck %trucktmp%; &else; ~
                &s truck'
%2 CHECKBOX ck2 RETURN '&if %ck2% &then; &s commodity %commoditytmp%; ~
                &else; &s commodity'
%3 CHECKBOX ck3 RETURN '&if %ck3% &then; &s states %statestmp%; ~
                &else; &s states'
%input1 INPUT trucktmp 20 ~
    TYPEIN NO SCROLL YES ROWS 4 FILE '$HTFSDIR/polygon/*.trk' -INFO
%input2 INPUT commoditytmp 20 ~
```

```
    TYPEIN NO SCROLL YES ROWS 4 FILE '$HTFSDIR/polygon/*.cdt' -INFO
%input3 INPUT statestmp 20 ~
    TYPEIN NO SCROLL YES ROWS 4 COVER '$HTFSDIR/polygon/*' -ALL
%button1 BUTTON 'ACCEPT' &s cancel .FALSE.; &if [NULL %truck%] &then; ~
          &s truck #; &if [NULL %commodity%] &then; &s commodity #; ~
          &if [NULL %states%] &then; &s states #; ~
          &thread &focus &on &other
%button2 BUTTON 'CANCEL' &s cancel .TRUE.; &thread &focus &on &other
%forminit &s truck; &s commodity; &s states; &s ck1 .FALSE.; ~
      &s ck2 .FALSE.; &s ck3 .FALSE.
```

Appendix D-9

```
7
/*************************************************************/
/**
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:   Prompts the user to select RAIL, OPWT, DEMAND, AND
/**          ECONOMIC databases...
/**
/** DATE:     8 MARCH 1995
/*************************************************************/

  ^Commodity Supply:

   %disp1

  ^Commodity Demand:

   %disp2

  ^Rail Cost:

   %disp3

 ================================================

      ^Select the Supply, Demand, and Rail Databases:

     %input1            ^Check one of each:

                     %ck1 Supply

                     %ck2 Demand

                     %ck3 Rail

             %button1   %button2

%disp1 DISPLAY supply 50
%disp2 DISPLAY demand 50
%disp3 DISPLAY rail 50
%input1 INPUT tmpdb 20 ~
    TYPEIN NO SCROLL YES ROWS 7 FILE '$HTFSDIR/srf/*.srf' ~
    CHARACTER
%ck1 CHECKBOX ck1 RETURN '&s supply %tmpdb%'
%ck2 CHECKBOX ck2 RETURN '&s demand %tmpdb%'
%ck3 CHECKBOX ck3 RETURN '&s rail %tmpdb%'
%button1 BUTTON 'ACCEPT' &s cancel .FALSE.; &thread &focus &on &other
%button2 BUTTON 'CANCEL' &s cancel .TRUE.;  &thread &focus &on &other
%formopt SETVARIABLES IMMEDIATE
%forminit &s ck1 .FALSE.; &s ck2 .FALSE.; &s ck3 .FALSE.; &s demand; ~
      &s supply; &s rail
```

Appendix D-10

```
7
/*************************************************************/
/**
/**          S T A _ T R K _ C D T . M E N U
/**
/** PROGRAMMER: Todd Raynor
/**
/** PURPOSE:    Allows the user to select an existing scenario ...
/**
/** DATE:       8 MARCH 1995
/**
/*************************************************************/

    ^Trucks Database:    %disp1

    ^Commodity Database: %disp2

    ^States Database:    %disp3

  ============================================================

     ^Select the Truck, Commodity, and State Databases to Extract:

  ^Trucks:          ^Commodity:          ^States

  %input1           %input2              %input3

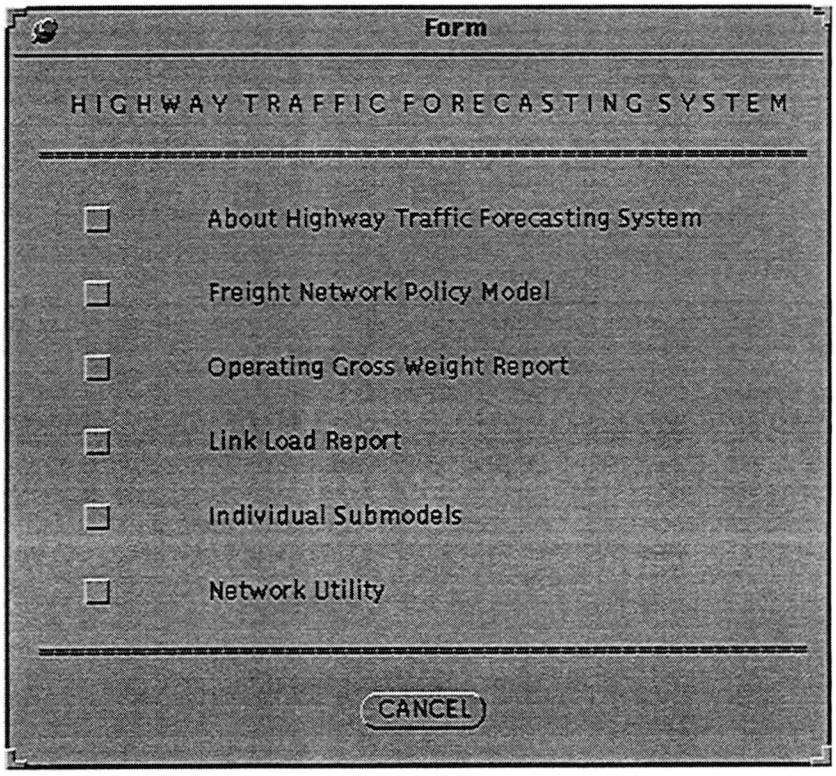



                %button1   %button2

%disp1 DISPLAY truck 50
%disp2 DISPLAY commodity 50
%disp3 DISPLAY state 50
%input1 INPUT truck 20 ~
    TYPEIN NO SCROLL YES ROWS 4 FILE '$HTFSDIR/srf/*.trk' ~
    CHARACTER
%input2 INPUT commodity 20 ~
    TYPEIN NO SCROLL YES ROWS 4 FILE '$HTFSDIR/srf/*.cdt' ~
    CHARACTER
%input3 INPUT state 20 ~
    TYPEIN NO SCROLL YES ROWS 4 FILE '$HTFSDIR/srf/*.sta' ~
    CHARACTER
%button1 BUTTON 'ACCEPT' &s cancel .FALSE.; &thread &focus &on &other
%button2 BUTTON 'CANCEL' &s cancel .TRUE.;  &thread &focus &on &other
```

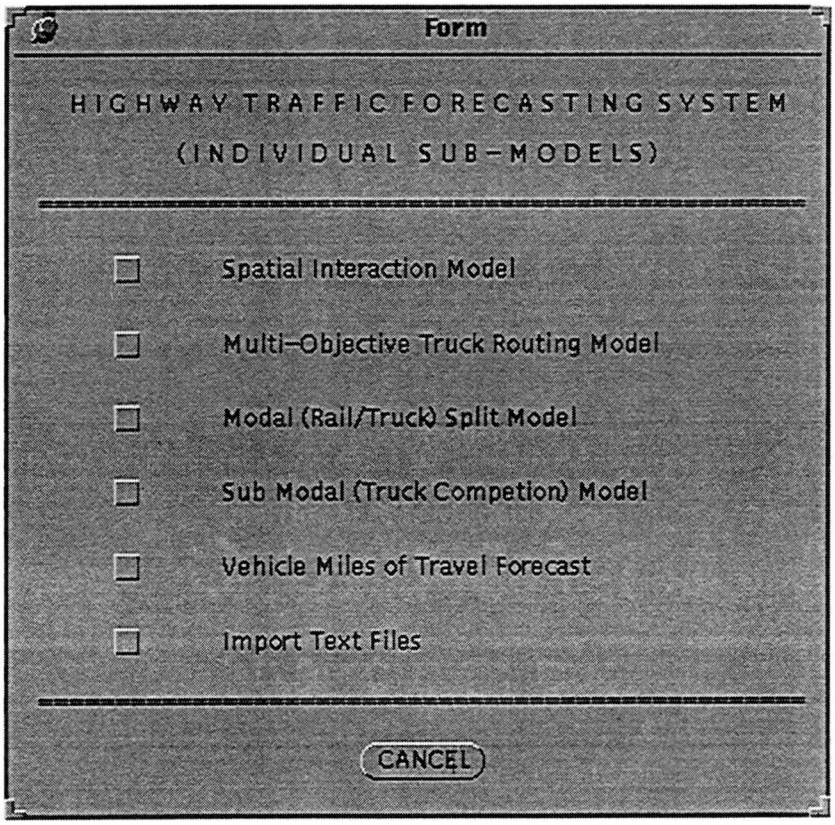
Appendix E-1

HTFS's Main Menu



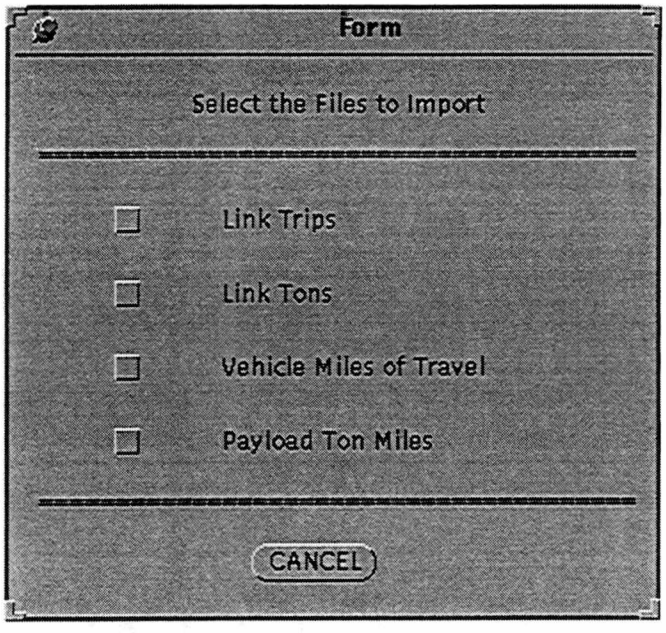
Appendix E-2

Individual Sub-Models Menu



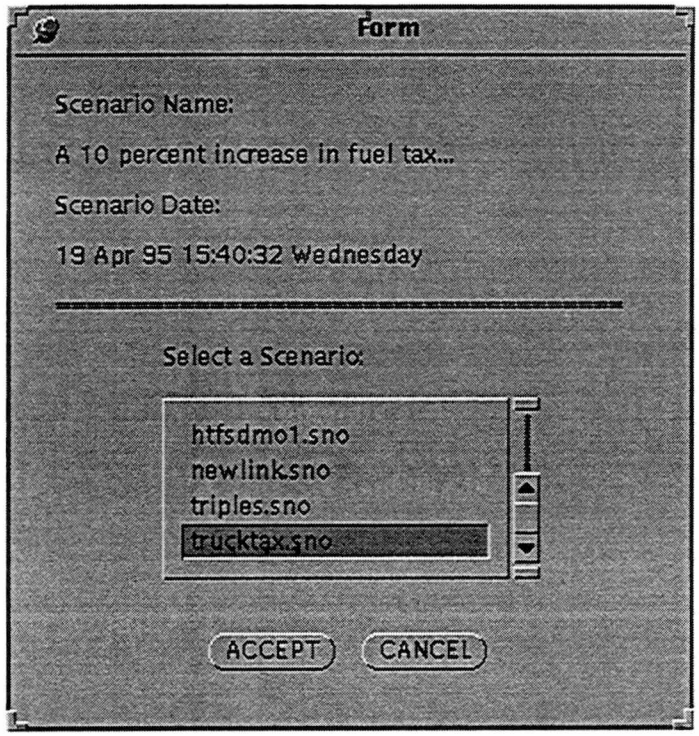
Appendix E-3

Import Menu



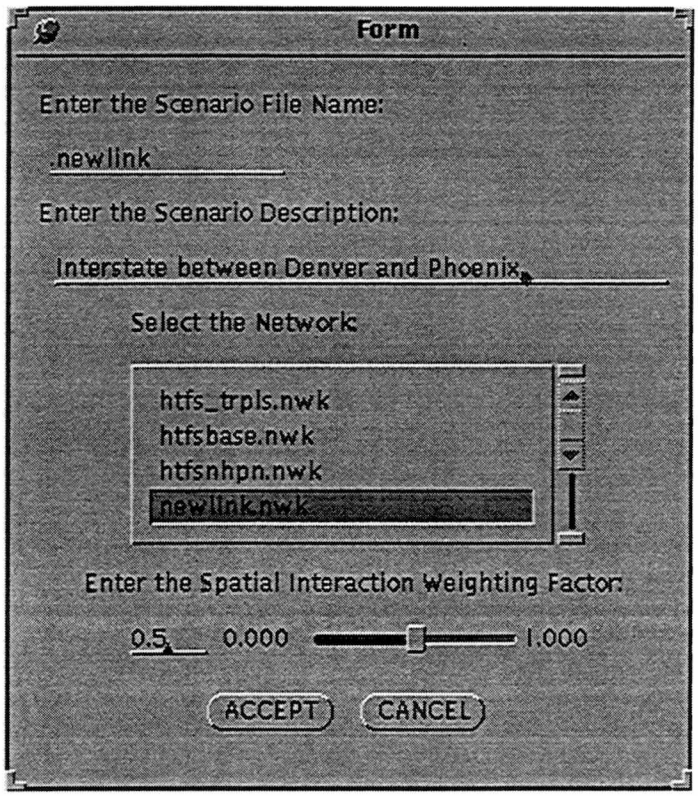
Appendix E-4

Existing Scenario's Menu



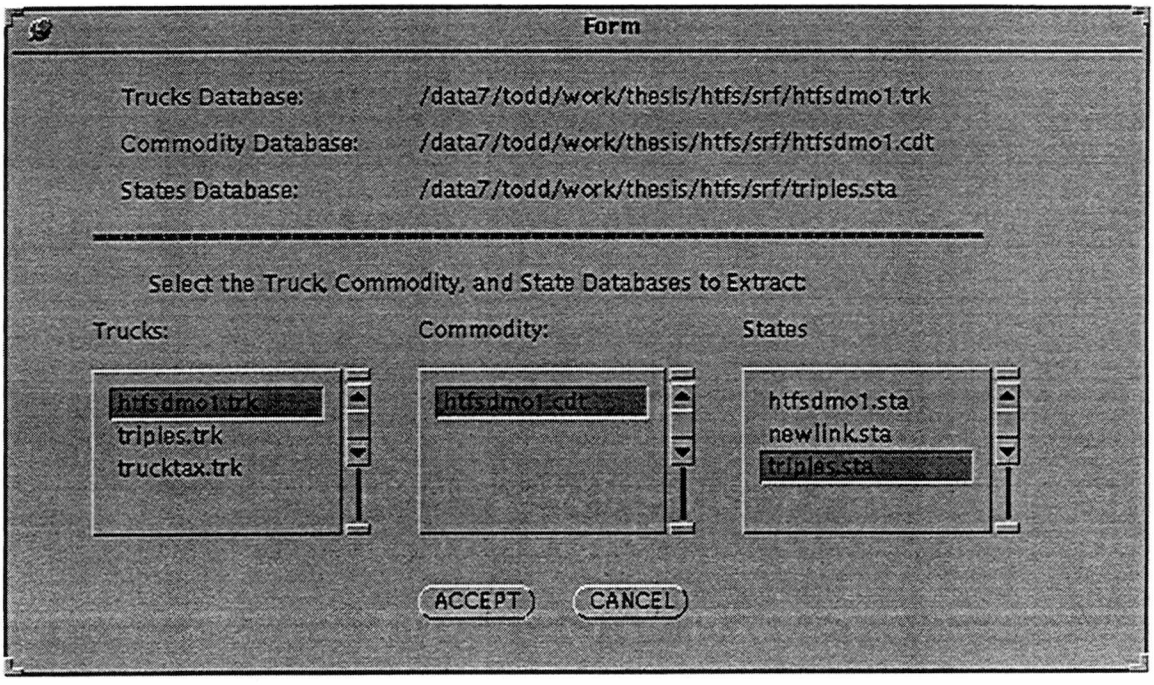
Appendix E-6

New Scenario Menu



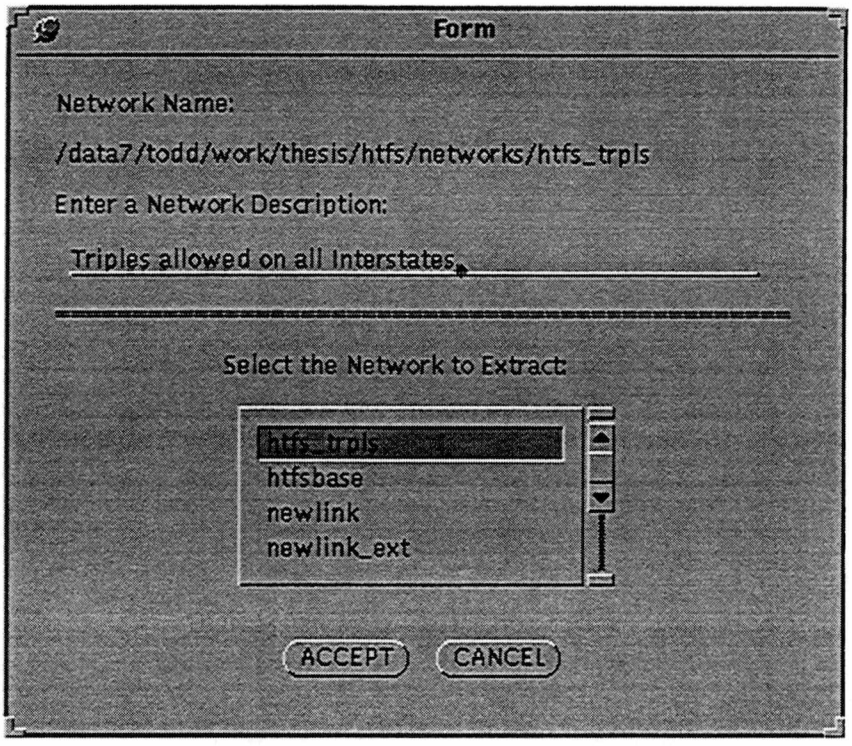
Appendix E-6

Truck, Commodity, and States Selection Menu



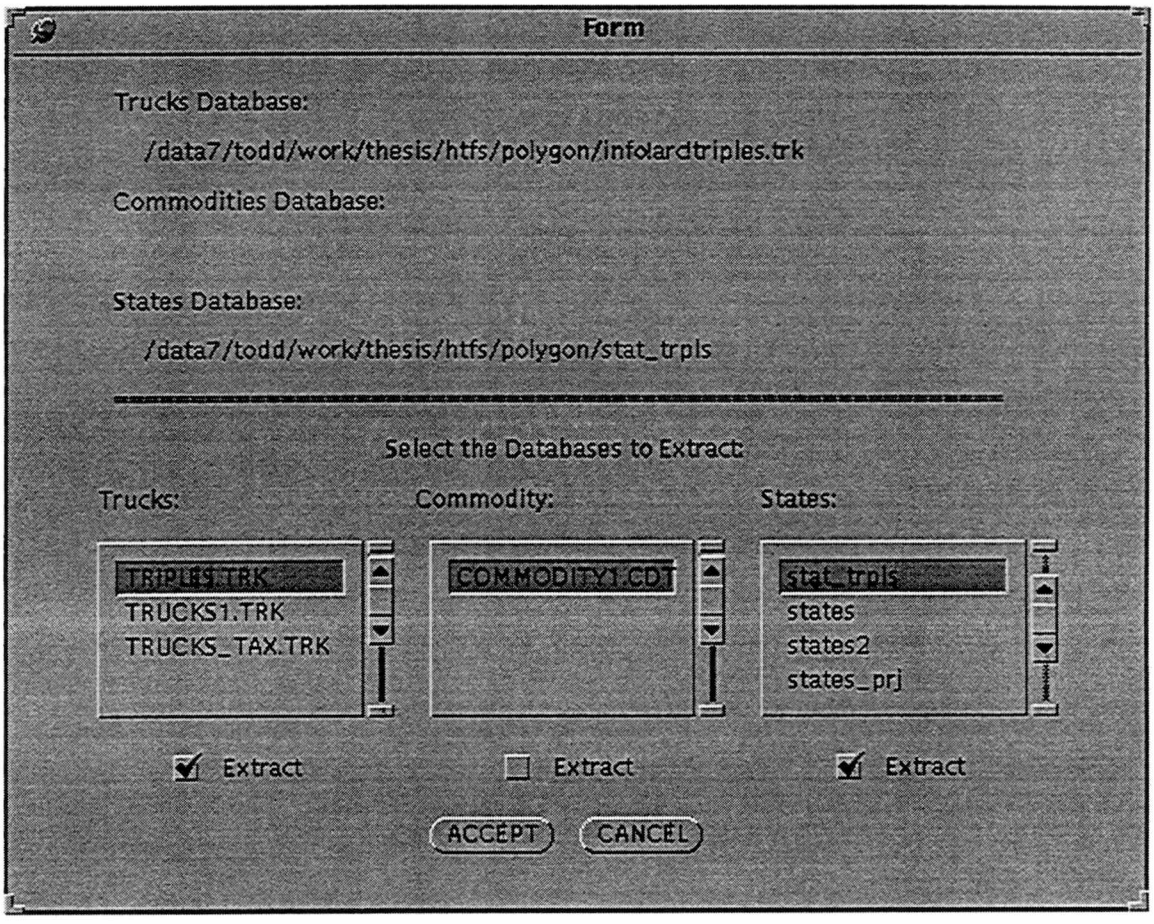
Appendix E-7

Network Extraction Menu



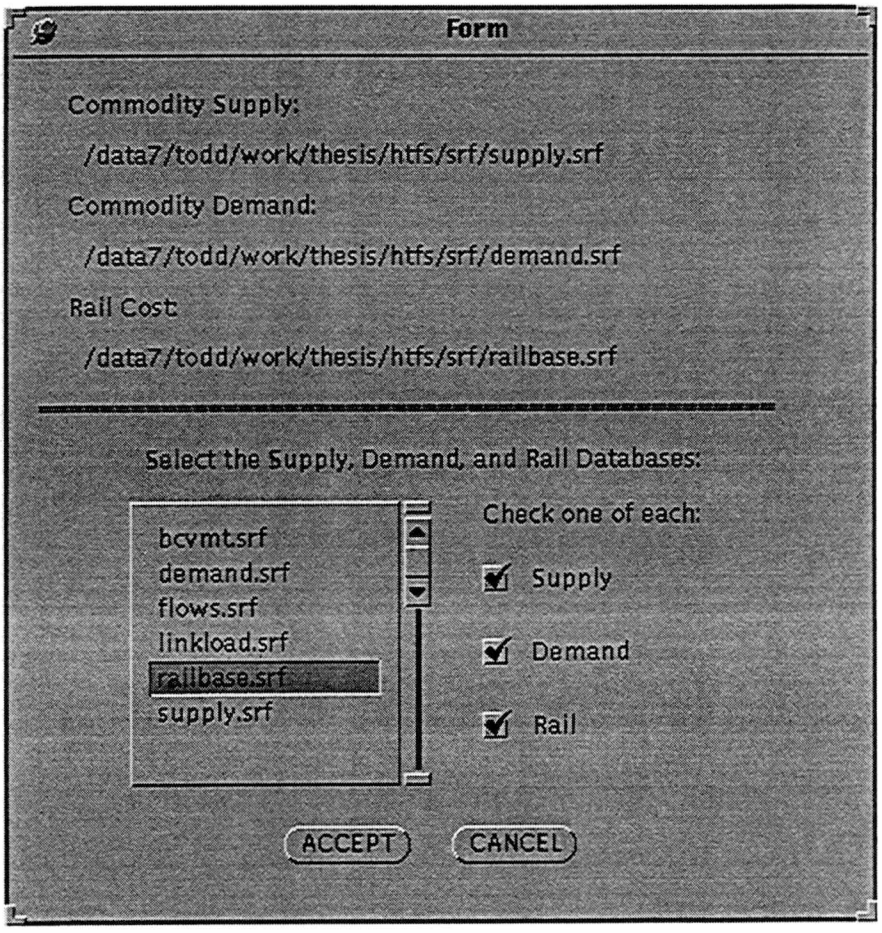
Appendix E-8

Truck Commodity, and State's Extraction Menu



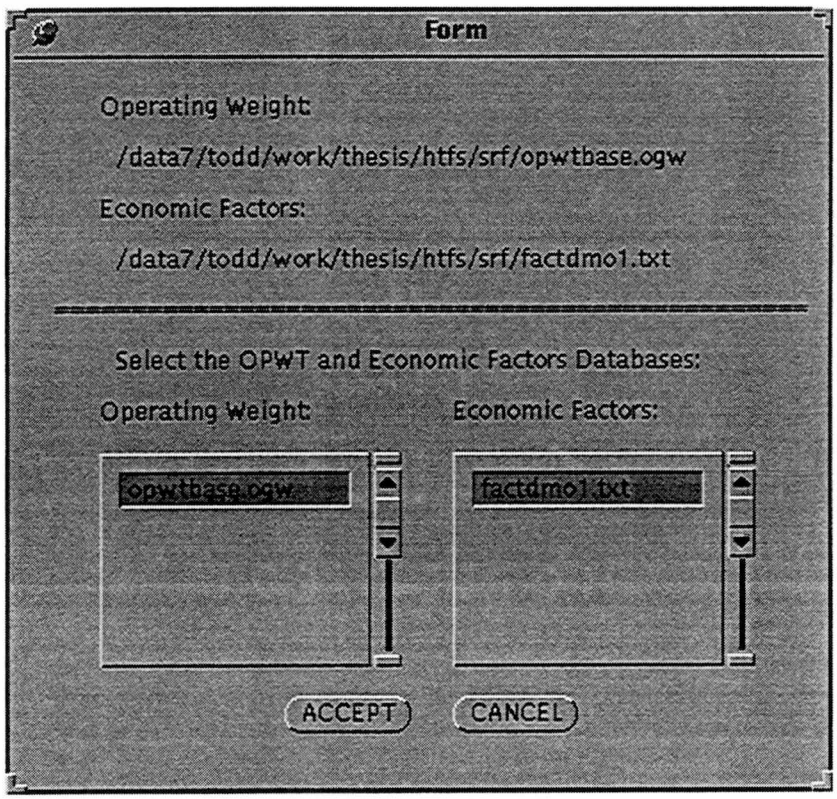
Appendix E-9

Supply, Demand, and Rail Cost Selection Menu



Appendix E-10

Operating Weight and Economic Factors' Selection Menu



Appendix F

Data Sources Used for the Basecase Scenario

The following data sources have been used to generate the input data found in the FNET "Basecase" scenario (HTFSDMO1.APP):

- o Truck and Rail Commodity Flow Data: Inter-BEA tonnage movements by 23 (twenty-three) 2-Digit SIC Commodity Group (or Group Combinations). This is proprietary data. It was obtained by FHWA from Reebie Associates, under the title "TRANSEARCH: Data Base of U.S. Freight Flows, 1987". The data was used to calibrate the HTFS spatial interaction models, where a clear distance decay effect was evident in tonnages moved.

- o Highway Network Data: The HTFS Highway Network is a reduced form of the ORNL National Highway Planning Network (NHPN). The reader is directed to the NHPN Technical Manual (Peterson, 1991) for further details. For the HTFS, the NHPN network database of some 55,000 links was reduced to a network composed of some 6,000 links, oriented towards inter-BEA flows. (The HTFS is capable of handling other, and larger networks, if required to do so).

- o Rail Operating Cost Data: Rail operating costs per ton-mile are provided at 200 mile travel distance intervals. These costs need to be updated. The costs provided with HTFS Version 2.0B are based on 1977 data.

- o Truck Operating Cost Data: Truck operating costs per mile, by each of 13 truck configurations were estimated using the cost data contained in the SYDEC Inc./Jack Faucett Associates report to FHWA, titled "The Effect of Size and Weight Limits on Truck Costs" (JACKFAU-90-352-1) dated June 1990. This report contains cost estimates for a range of gross vehicle weights and body types for a large number of truck configurations. To select representative costs based upon these values, the 1987 Truck Inventory and Use Survey (TIUS) was consulted, and costs selected on the basis of the most common body and operating weight type. (Given the variability possible in this cost data, and the resources available, a rigorous statistical exercise to obtain average costs by configuration, across all operating weights and body types was not carried out).

- o Operating Gross Vehicle Weight (OGW) Distributions by each of the 13 truck configurations analyzed within HTFS: this data came from the 1985 Truck Size and Weight (TS&W) truck count study.

- o Existing State specific maximum allowable OGW limits were obtained from FHWA.

The VMT and associated PTM Reconciliation Procedure applied to the FNET projections makes use of the following data sources:

- o The 1987 Truck Inventory and Use Survey (TIUS): to estimate the percentage of each truck configuration's VMT that is likely to be long haul (i.e. over 150 miles).

- o FHWA and ORNL modified versions of the VM-1 and VM-2 vehicle miles of travel tables published in "Highway Statistics"

- o The 1985 Truck Size and Weight (TS&W) survey of truck counts.

Appendix G-1

HTFS's Commodity Types

Commodity Identifier	Commodity
1	Farm, Fish & Marine Products
2	Metallic Ores
3	Coal
4	Nonmetallic Minerals
5	Food, Tobacco & Kindred Products
6	Textile, Apparel & Leather Products
7	Forest, Lumber & Wood Products
8	Furniture Or Fixtures
9	Pulp, Paper Or Allied Products
10	Printed Matter
11	Chemicals Or Allied Products
12	Petroleum, Natural Gas & Coal
13	Rubber Or Misc Plastics
14	Clay, Concrete, Glass Or Stone
15	Primary Metal Products
16	Ordnance & Fabricated Metal
17	Machinery
18	Electrical Equipment
19	Transportation Equipment
20	Instrum, Photo, Optical Equipmenmt
21	Misc Mixed Shipments
22	Waste Or Scrap Materials
23	Other Freight

Appendix G-2

Supply and Demand for Farm Fish and Marine Products

BEA Region	Supply	Demand
0	592497	62151
1	24808	274633
2	0	332744
3	38235	1075385
4	1388	164008
5	14086	688864
6	222212	820805
7	73885	351641
8	304590	135249
9	237227	1065126
10	22425	146440
11	310791	1850931
12	5571	103678
13	2672	153573
14	3257	69797
15	11481	334562
16	23739	1011069
17	374238	1393040
18	222436	2666253
19	82021	537906
20	20644	844061
21	75026	142682
22	264533	2011623
23	525421	278662
24	29454	267475
25	94618	371572
26	63598	1024022
27	19818	854910
28	37469	1852174
29	54636	75452
30	78428	289348
31	132188	429294
32	162668	185785
33	28874	231471
34	131381	214094
35	21822	3366925
36	48930	95264
37	296802	161436
38	226793	728169
39	391865	297758
40	455211	1088889
41	515242	256039
42	1270800	277150
43	975600	738905
44	85114	6799
45	9942	100398
46	240443	2605097
47	188381	507979
48	24364	1438975
49	130025	1143884
50	308106	1492748
51	1264	157529
52	20099	989334
53	345292	685803
54	933968	2460754
55	284632	8099

56	510988	920843
57	35511	163833
58	5442	170157
59	7	56090
60	0	43027
61	490	16070
62	60	46322
63	1989	96128
64	811719	566189
65	1967663	271766
66	816667	769731
67	1088595	350964
68	798857	21545
69	2149225	567814
70	524617	513124
71	1190690	142776
72	313498	217643
73	583148	236485
74	373805	96056
75	1141742	83660
76	1107285	35715
77	1027928	36536
78	1177220	439166
79	2330422	415951
80	1031537	18327
81	1076603	138323
82	4846916	4894728
83	3408301	293162
84	459024	2237151
85	0	32808
86	494922	1031582
87	229428	65257
88	253643	1299750
89	90651	60809
90	8067	580071
91	13117	26030
92	359971	50893
93	123504	581698
94	598118	3858243
95	4598434	6063866
96	313357	370842
97	32922	1277274
98	9991	4506275
99	460345	2501733
100	3631179	111467
101	7486776	121646
102	3717676	114091
103	2164872	994091
104	10455715	8826468
105	264432	119088
106	1809875	3570663
107	383686	508791
108	6345	818696
109	7253	1159471
110	156063	938518
111	109775	1093275
112	483537	4566081
113	84474	792454
114	7788	69137
115	627	72045
116	8181	145278
117	134019	49076
118	83623	721571
119	27015	523835

120	12306	3302463
121	358023	12805613
122	97332	11828
123	313728	253654
124	1947617	4471326
125	284472	41484
126	729100	127685
127	65880	17349
128	180001	2154709
129	418768	899340
130	547783	369192
131	3478	41621
132	136639	976141
133	1111188	834463
134	2543306	2416036
135	405989	27605
136	2885253	1059510
137	90908	583774
138	5476404	810725
139	4665576	209912
140	2207639	1119930
141	3550502	586296
142	6744779	2264789
143	7743310	255067
144	1056147	355964
145	838072	31321
146	4620605	24337
147	1708271	6806
148	5000156	549288
149	5258032	181404
150	926327	9875
151	3724257	18713
152	2823537	140562
153	121204	37540
154	681219	100789
155	108724	39902
156	2575329	2041912
157	1267016	69578
158	84903	41046
159	9400	130158
160	222395	74515
161	944026	1187570
162	2120	75833
163	79602	41848
164	421861	946559
165	3106180	369298
166	575726	546647
167	1197877	218996
168	1068551	83261
169	1697731	111591
170	34567	5558961
171	435804	12462536
172	212645	36031
173	270929	33306
174	6759	9046
175	2442675	1148237
176	477574	306689
177	1010634	2030440
178	3546372	1237296
179	2663143	4620436
180	1507973	779292

VITA

Todd Christopher Raynor was born October 30, 1964, in Visalia, California. He graduated from San Diego State University in 1993 with a Bachelor of Arts Degree in Geography. Upon graduation, he decided to pursue graduate work in Geography of Transportation and Geographic Information Systems at the University of Tennessee, Knoxville. In August, 1993, he began working full time for the University of Tennessee's Transportation Center as a Research Associate in the Geographic Information Systems Group.