GIS PROCESS OVERVIEW for PHASE 2 of the RAILROAD LINE-HAUL CAPACITY PROJECT

Geographic Information Systems Team
Tennessee Valley Authority
Norris, Tennessee

March 1998

Contributing Norris GIS Team Members:

GIS Project Leader:	Pamela J. Nabors
GIS Senior Technician:	Garry E. Chappelle
GIS Technician:	Veronica A. Long
GIS Technician:	Larry E. Shelton
GIS Technician:	Deborah K. Smith
Arc/Info Consultant:	David A. Hankins

with

Navigation Team GIS Specialist:	M. Carolyn Koroa
Navigation Team GIS Intern:	Jim W. Cunningham

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AML - Arc Macro Language

BTS - Bureau of Transportation Statistics

FTP - File Transfer Protocol

GIS - Geographic Information System

NTAD - National Transportation Atlas Databases

INTRODUCTION

In late June 1997, the TVA Navigation Team employed members of the TVA Norris GIS Team to conduct the second phase of a research and development project for determining the line-haul capacity of selected railroad lines in the United States. The objective was to use a Geographic Information System to simulate routing railroad shipments over a digital line network and produce a list of specialized route identification numbers for the customer. An overview of the process is graphically depicted in Figure 1. The input data, processes, and output data are discussed further in the following sections.

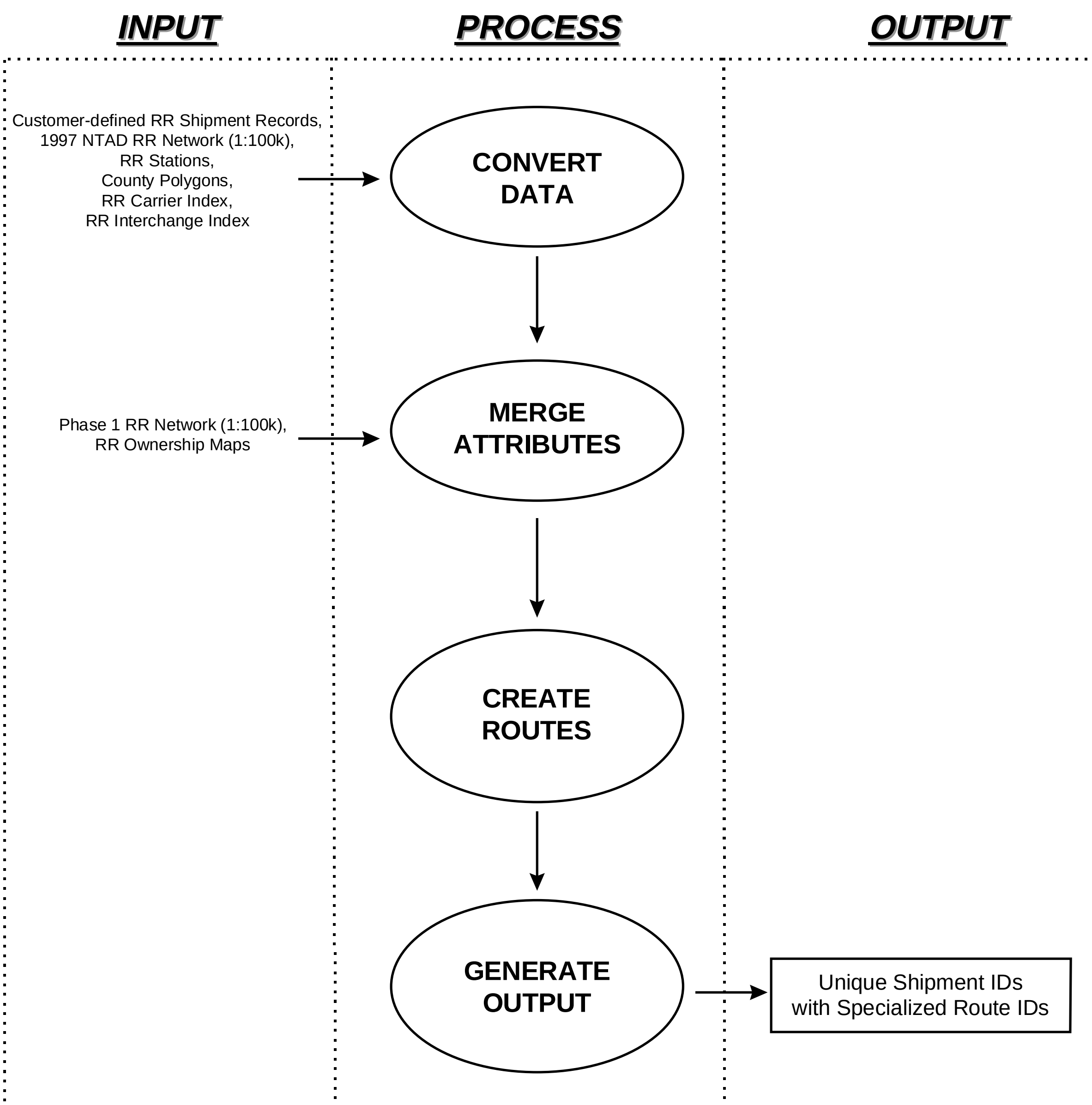


Figure 1. GIS Process Overview for Phase 2.

INPUT DATA

There were eight input data sets used for the project (as shown in Figure 1):

1). Customer-defined railroad shipment records. The customer originally sent about 500,000 shipment records to be routed. These records were generated from the 1995 Carload Waybill Sample and represented 2-3% of all railroad movements for that year. Since a separate record existed for each *type* of shipment (coal, corn, etc.), many of these shipment records had the same route (i.e., same origin, destination, and railroad owner). Therefore, after we discovered the large amount of time required to route so many shipments using Arc/Info, Dr. Burton combined duplicate shipment routes and generated a unique identifier for each group. The new, pared shipment data set contained about 75,000 records with the following attributes:

FIELD NAME	*DESCRIPTION*
UNIQUE	Unique Shipment Identifier Assigned by Dr. Mark Burton
OFSAC	Originating Station FSAC Code
ORR	Originating Railroad American Association of Railroads Number (AARNO)
INT1	First Interchange Location Alpha Code
RR2	Second Railroad American Association of Railroads Number
INT2	Second Interchange Location Alpha Code
RR3	Third Railroad American Association of Railroads Number
INT3	Third Interchange Location Alpha Code
RR4	Fourth Railroad American Association of Railroads Number
INT4	Fourth Interchange Location Alpha Code
TRR	Terminating Railroad American Association of Railroads Number
TFSAC	Terminating Station FSAC Code
NUMRR	Number of Shipment Segments
OFIP	Originating County FIPS Code
TFIP	Terminating County FIPS Code

2). 1997 NTAD 1:100,000 scale U. S. railroad network (see website http://www.bts.gov). A pre-release version was acquired through the Department of Transportation's Bureau of Transportation Statistics and used as the underlying topology for the project. Refer to the rail100k.met metadata file on the 1997 NTAD compact disc for more details.

3). Railroad station data purchased from Alber Leland, Inc. A completed data set was not available at the beginning of the project, so a preliminary copy of the data was delivered in August 1997. An updated preliminary version was delivered again in October and used as the final data set. Station data contained the following coordinate information (from the RCOORUS file):

FIELD NAME	*DESCRIPTION*
STATION_ID	Unique Station Identifier
LATITUDE	Latitude of Railroad Station in Decimal Degrees
LONGITUDE	Longitude of Railroad Station in Decimal Degrees

and attribute information (from the RAILUS file):

FIELD NAME	*DESCRIPTION*
STATION_ID	Unique Station Identifier
STAT_NAME	Name of Railroad Station
STAT_STATE	State Name of Railroad Station
STAT_COUNT	County Name of Railroad Station
FSAC	Freight Station Accounting Code, *Corresponds with Shipment Record's* OFSAC, TFSAC *Attributes*
OPSL	Open and Prepaid Station List Number
SPLC	Standard Point Location Code
ZIPCODE	Rating Zip Code
SCAC	Serving Carrier Standard Carrier Alpha Code

4). County polygons. The county shapefiles on the "ESRI Data & Maps, Volume 1" compact disc provided with ESRI's ArcView 3.0 were used to provide county polygon data with the following information:

FIELD NAME	*DESCRIPTION*
NAME	Name of County
STATE_NAME	State Name of Residing County
FIPS	Full County FIPS Code, *Corresponds with Shipment Record's* OFIP, TFIP *Attributes*

5). Railroad carrier index provided by the Navigation Team. This index was created to provide a link between the customer's shipment records and the Alber Leland station records via the given carrier information. To do this, a list was first generated of all the American Association of Railroads numbers (AARNO) occurring in the shipment records (ORR, RR2, RR3, RR4, TRR). Carrier name alpha codes (ALPHA) were then added for each AARNO using the Official Railway Guide as a reference.

FIELD NAME	*DESCRIPTION*
ALPHA	Railroad Carrier Alpha Code (carrier name abbreviation), *Corresponds with Station List's* SCAC *Attribute*
CARRIER_NAME	Full Name of Railroad Carrier
AARNO	American Association of Railroads Number, *Corresponds with Shipment Record's* ORR, RR2, RR3, RR4, TRR *Attributes*

6). Railroad interchange index provided by the Navigation Team. This index was created to provide a link between the customer's shipment records and the Alber Leland station records for interchange points. First, a list was generated of all the interchange codes (INT_CODE) occurring in the shipment records (INT1, INT2, INT3, INT4). Corresponding interchange names and state names (INTERCHANGE, INT_STATE) were then added using the Open and Prepaid Station List as a reference.

FIELD NAME	*DESCRIPTION*
INT_CODE	Interchange Alpha Code (interchange name abbreviation), *Corresponds with Shipment Record's* INT1, INT2, INT3, INT4 *Attributes*
INTERCHANGE	Interchange Full Name, *Corresponds with Station List's* STAT_NAME *Attribute*
INT_STATE	State Name of Residing Interchange, *Corresponds with Station List's* STAT_STATE *Attribute*

7). Specialized 1995 NTAD railroad network (1:100,000 scale) with Phase 1 attributes. The specialized route identification numbers (ROUTEID field) and railroad ownership attributes from the Phase 1 network were reused in Phase 2. Ownership attributes from Phase 1 included the OWNER field from the 1995 NTAD railroad network (1:100,000 scale), and the RROWN1, RROWN2, RROWN3 fields from 1996 NTAD railroad network (1:2,000,000 scale). Refer to the Phase 1 documentation (May 1997) for a more detailed description of the Phase 1 attribute data.

8). Ownership information. Various paper maps produced by individual railroad carriers were used to add ownership attributes when necessary.

PROCESS OVERVIEW

Phase 2 of the GIS railroad line-haul capacity project was conducted using Arc/Info 7.0.4 and ArcView 3.0 running on a network of Sun workstations and Pentium PCs. The following sections describe how the input data were converted, attributes were merged, routes were created, and output was generated.

CONVERT DATA

1). Railroad shipment records were converted from an ASCII columnar format to INFO database format using an AML macro. The macro used the Tables module *DEFINE*[1] command, and the Info module *SEL* command and *GET* command (with the *COPY* and *ASCII* options). The *CHANGE* command from the Tables module of Arc/Info was then used to strip trailing blanks from the interchange fields (INT1, INT2, INT3, INT4).

2). The 1997 NTAD 1:100,000 scale railroad network was converted using the BTS bts2arc.aml conversion macro (see website http://www.bts.gov/gis/ntatlas/btsarc.aml).

3). The railroad station coordinate data purchased from Alber Leland, Inc. was converted by using the Arc/Info *GENERATE* command. The **station attribute data** was received with double quotes around each item, so all data were imported into Info as character fields, then the FSAC field was converted to integer and divided by 100.

[1] Arc/Info commands are capitalized and italicized in this document.

4). **U.S. county polygon shapefiles** from the “ESRI Data & Maps, Volume 1” compact disc were copied to a UNIX hard drive and converted to an Arc/Info coverage using the following commands: *SHAPEARC*, *CLEAN*, *REGIONPOLY*.

5). **The railroad carrier index** provided by the Navigation Team was exported from MicroSoft Excel into dBASE IV format and copied to a UNIX hard drive. The data were then converted to Info format using the *DBASEINFO* command.

6). **The railroad interchange** index provided by the Navigation Team was exported from MicroSoft Excel into dBASE IV format and copied to a UNIX hard drive. The *DBASEINFO* command was used to convert the data to Info format.

7). No conversion was necessary for the **Phase 1 network**.

8). No conversion was necessary for using the paper **ownership maps**.

MERGE ATTRIBUTES

Assemble Network Attributes

Attributes from the railroad network used in Phase 1 of this project were transferred to the new Phase 2 network (1997 NTAD 1:100,000 scale) using the *NEAR* and *JOINITEM* Arc/Info commands. A visual check of the network was made along with any necessary manual corrections, especially for the Phase 1 specialized route numbers (ROUTEID field). Only the ROUTEID, ownership, and state FIPS attributes were preserved on the new network. Therefore, the new network attributes were:

FIELD NAME	*DESCRIPTION*
STFIPS	State FIPS Code
RROWNER	1997 NTAD 1:100k Railroad Owner Name Abbreviation
ROUTEID	Dr. Mark Burton’s Specialized Route Identification Number
OWNER	1995 NTAD 1:100k Railroad Owner Name Abbreviation
RROWN1	1996 NTAD 1:2mill First Railroad Owner Name Abbreviation
RROWN2	1996 NTAD 1:2mill Second Railroad Owner Name Abbreviation
RROWN3	1996 NTAD 1:2mill Third Railroad Owner Name Abbreviation

The 1997 NTAD 1:100,000 scale network did not have adequate ownership information, so ownership information from the Phase 1 network was combined with it to produce a new data field: COMBO_OWN, and the other ownership fields were dropped. Ownership was assigned in the following priority to emulate actual 1995 ownership as close as possible:

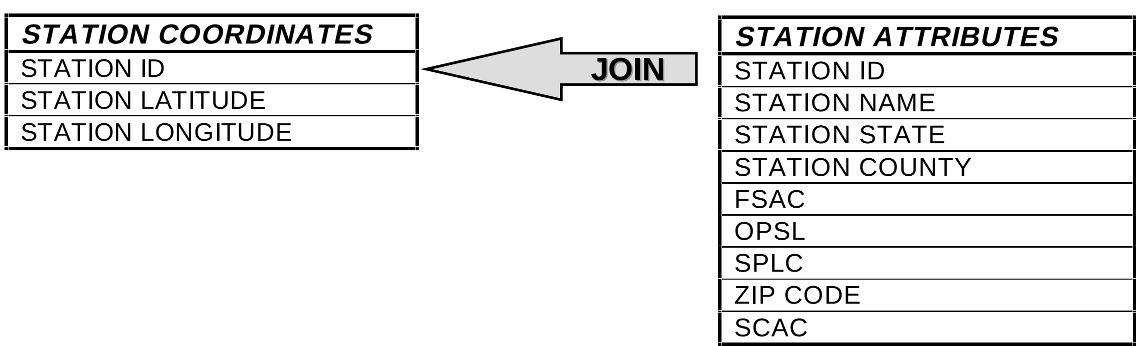
1.) 1996 NTAD 1:2,000,000 scale ownership attributes (RROWN1, RROWN2, RROWN3),
2.) 1995 NTAD 1:100,000 scale ownership attributes (OWNER), then
3.) 1997 NTAD 1:100,000 scale attributes (RROWNER).

Even after combining all ownership fields, only about 60% of the arcs had ownership attributes. So, the network was transferred to the Navigation Team GIS specialist and intern who used various paper maps produced by individual railroad carriers to manually enter additional ownership information. To save time during the editing process, the Phase 2 railroad network was divided into 2 parts (eastern and western U.S.) and worked on simultaneously. The western portion was *EXPORT*ed and FTPed to the Navigation Team UNIX workstation and edited with Arc/Info. The eastern portion was converted via *ARCSHAPE*[2] and transferred to their PC and edited with ArcView 3.0a. Upon completion of their manual edits, the network was transferred back to the Norris GIS Team. The eastern network was converted back to a UNIX coverage using *SHAPEARC*, and the western network was *IMPORT*ed. After *APPEND*ing the eastern and western portions back together, the network was spot checked for topological and attribute errors. Two more data fields were then added for calculating and displaying routes. Therefore, the final railroad network contained the following arc attributes:

FIELD NAME	*DESCRIPTION*
STFIPS	State FIPS Code
ROUTEID	Dr. Mark Burton's Route Identification Number
COMBO_OWN	Ownership Alpha Code - compiled from multiple sources
IMPEDE	Impedance value for calculating a route
IMPEDESYM	Arc/Info drawing symbol code

Assemble Station Attributes

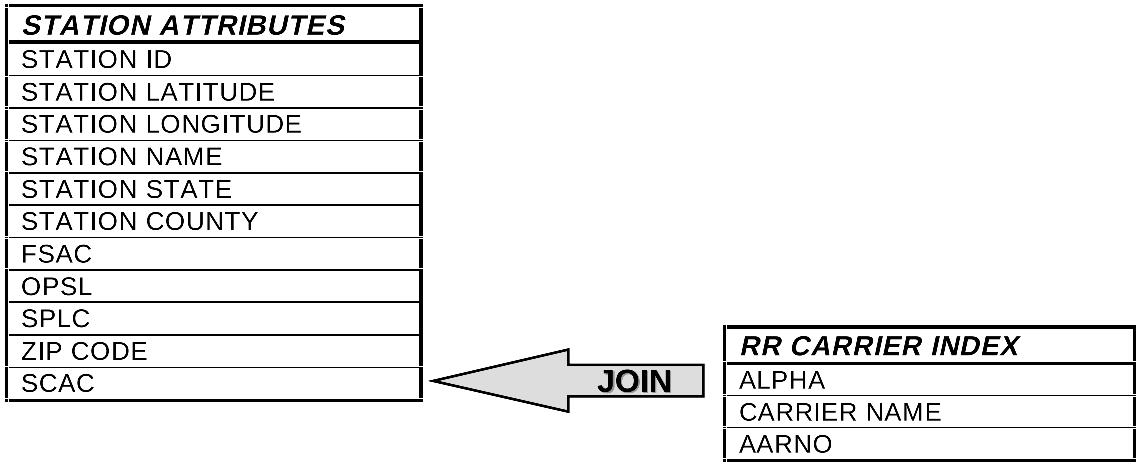
After the Alber Leland, Inc. railroad station coordinate data was converted to an Arc/Info point coverage, the station attribute data was converted and joined to it via the *JOINITEM* command using the unique station identification numbers as the key field (depicted below).



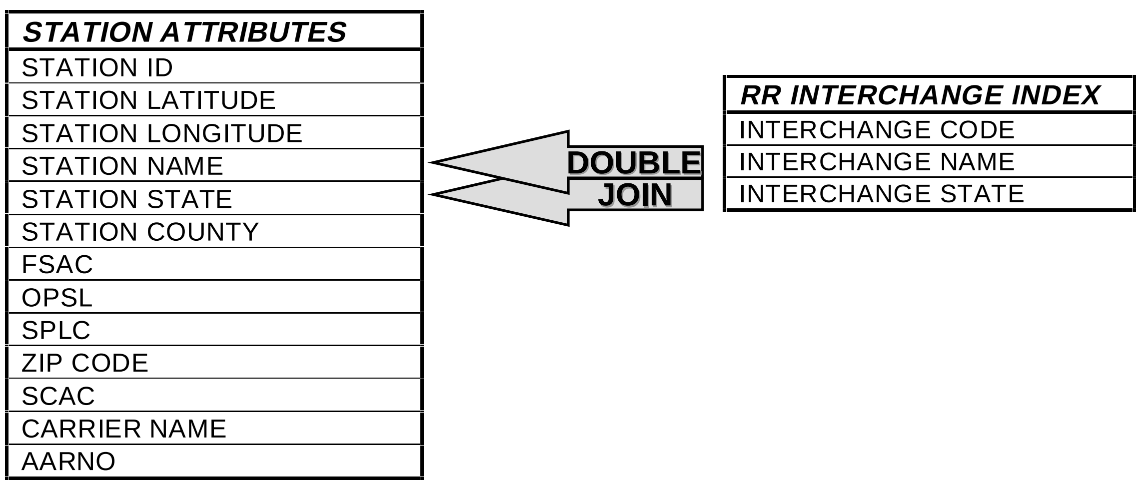
The carrier index was then joined to the station data via railroad alpha codes as shown below.

---

[2] It was later discovered that this caused a problem with the route identification numbers. The ROUTEID field from the Phase 1 network was defined as a Numeric field with an internal width of 4. The *ARCSHAPE* command forced a decimal point in the output text file, therefore truncating all numbers to three digits. Although the routes had to be processed again, it provided an opportunity to make enhancements to the whole process and its final products.



Next, the *REDEFINE* and *JOINITEM* commands were used to join the interchange index to the station data. The interchange name and state fields were joined with the station name and state fields via a double join as depicted below.



A link was then created between the station data and the railroad network data. Each station was assigned the internal address of the nearest node on the railroad network using the following Arc/Info commands:

- *BUILD* - to create node topology for the rail network,
- *NEAR* - to assign each station the nearest railroad network internal node number (PAREDRAIL#) and the distance between nodes (DISTANCE),
- *JOINITEM* - to join the railroad network node attribute table to obtain the railroad node's user identification number (PAREDRAIL-ID) to be used by the routing program. This also included the railroad network arc the node is associated with (ARC#).

The attributes of the final station data are listed below.

FIELD NAME	*DESCRIPTION*
STATION_ID	Unique Station Identifier
LATITUDE	Station's Latitude in Decimal Degrees
LONGITUDE	Station's Longitude in Decimal Degrees
STAT_NAME	Name of Station
STAT_STATE	State Name of Station
STAT_COUNT	County Name of Station
FSAC	Freight Station Accounting Code, *Corresponds with Shipment Record's* OFSAC, TFSAC *Attributes*
OPSL	Open and Prepaid Station List Number
SPLC	Standard Point Location Code
ZIPCODE	Rating Zip Code
SCAC	Serving Carrier Standard Carrier Alpha Code
CARRIER_NAME	Name of Railroad Carrier
AARNO	American Association of Railroads Number, *Corresponds with Shipment Record's* ORR, RR2, RR3, RR4, TRR *Attributes*
INT_CODE	Alpha Code (interchange name abbreviation), *Corresponds with Shipment Record's* INT1, INT2, INT3, INT4 *Attributes*
PAREDRAIL#	Link to Nearest Railroad Network Node
DISTANCE	Distance to Nearest Railroad Node
ARC#	Internal Identification Number of Associated Railroad Arc
PAREDRAIL-ID	User Identification Number of Nearest Node

Assemble Alternate Station Point Attributes

It was necessary to create an alternate data set to use when a railroad station point was not found in the Alber Leland data set. Therefore, an Arc/Info point data layer was created from railroad network nodes. County FIPS and ownership attributes were added so that origin and destination points could be selected via the customer's shipment record data (ORR, OFIP, TRR, TFIP). The following is a list of main commands used to create the alternate data layer:

- *NODEPOINT* - to create a new point coverage from the nodes in the railroad network,
- *RELATE* - to copy the ARC# values from the rail network nodes, and COMBO_OWN values from the rail network arcs to the new point coverage,
- *JOINITEM* - to join the carrier index attributes to the new point coverage,
- *IDENTITY* - to join the county attributes to the new point coverage,
- *ALTER* - to change field descriptions.

The final alternate data set attributes were:

FIELD NAME	*DESCRIPTION*
RAILPOINTS-ID	Arc/Info User Point Identification Number
Alternate Field Name:	PAREDRAIL-ID
ARC#	Internal Identification Number of Associated Railroad Arc
COMBO_OWN	Ownership Alpha Code - compiled from multiple sources
AARNO	American Association of Railroads Number, *Corresponds with Shipment Record's* ORR, RR2, RR3, RR4, TRR *Attributes*
COUNTY_NAME	Name of County where Point is Located
Previous Field Name:	NAME
STATE_NAME	Name of State where Point is Located
FIPS	Full County FIPS Code, *Corresponds with Shipment Record's* OFIP, TFIP *Attributes*

CREATE ROUTES

Once the data were prepared, the next step was to create the shipment routes. Three AML macros were produced to accomplish this. The main macro (AutoRoute.AML) was created to loop through the shipment records, call the necessary routines to process them (including the external routines ImpedeMany.AML and RouteBills.AML), and create the output files. Each AML is discussed further below.

Computer processing time was extensive due to the tremendous amount of data and the complexity of calculations. Therefore, the GIS Team divided the shipment records into batches and used as many central processing units and hard drives as possible in parallel. Originally, 12 CPUs were used with 14 different hard drives, but the maximum number of Arc/Info Network module licenses was 5, so the number of batches running simultaneously were reduced. Only 5 to 7 of the Sun Ultra workstations were used at one time on as many local hard drives as possible. Even after much of the GIS Team's computer network was upgraded to 100 megabyte Ethernet lines and two 9 megabyte hard drives were purchased by the Navigation Team, the final round of processing took approximately 4 weeks to process the 75,000 shipment records.

Main AML

The main AML macro used CURSOR commands to loop through the customer's shipment records, one shipment leg at a time. For each leg of a shipment, an attempt was made to find an originating and terminating node based on the following logic.

If the shipment route does not have any interchange points (only one leg):

- Use the shipment record originating and terminating FSAC codes and railroad owner AAR numbers (OFSAC, ORR, TFSAC and TRR fields) to find matching origin and destination points in the railroad station file (via the FSAC and AARNO fields). Store the identification number of the nearest nodes on the railroad network (PAREDRAIL-ID fields) in two variables, namely *from_node* and *to_node*, to pass on to the routing AML.

- If an origin or destination point cannot be found, use the shipment record county FIPS code (OFIP or TFIP) to find a matching point in the alternate station file for the current owner (ORR or TRR).
- If an origin or destination point still cannot be found, then use the shipment record county FIPS code (OFIP or TFIP) to find a point on the network within the county (via the alternate station file), regardless of the owner.
- If no match can be established[3], write the shipment record number to an error file.

If the shipment route has interchange points:

- For the first leg of the route, use the shipment record originating FSAC code and railroad owner abbreviation (OFSAC, ORR fields) to find a matching origin point in the railroad station file. If no match was found in the station file, then use the alternate station file as stated above. If no match can be established, write the shipment record number and leg number to an error file. Otherwise, store the identification number of the nearest node on the railroad network (PAREDRAIL-ID field) in a variable (*from_node*) to pass on to the routing AML.
- Find an interchange point by matching the shipment record interchange code and railroad owner AAR number (for example, INT1 and RR2 fields) to a point in the railroad station file (via the INT_CODE and AARNO fields). If an interchange point cannot be found for that owner, then find a matching point with the same interchange abbreviation, regardless of the owner. If no match can be established, write the shipment record number and leg number to an error file. Otherwise, store the node identification number in the *to_node* variable.
- For each subsequent leg, copy the *to_node* value to the *from_node* variable and find the next interchange point (such as INT2, RR3) until reaching the last leg.
- For the last leg of the route, use the shipment record terminating FSAC code (TFSAC) and railroad owner abbreviation (TRR) to find a matching destination point in the railroad station file. If no match was found in the station file, then use the alternate station file as stated above. If no match can be established, write the shipment record number and leg number to an error file. Otherwise, store the identification number in the *to_node* variable to pass on to the routing AML.

Once the *to_node* and *from_node* variables were established, and were not equal to eachother, then the impedance values for the current owner were set on the network by calling the external impedance routine (only if the ownership had changed since the previous shipment leg). Afterward, the route was created for that leg via the external routing routine, and output was generated.

Impedance AML

The impedance AML macro set impedance values on the railroad network by assigning numbers to each arc's IMPEDE field via Arc/Info's Tables module. Since the routing algorithm used the shortest path method, impedance values were based on arc length (i.e., travel distance). The higher the number, the more difficult it was to travel across the arc (i.e., portion of track). The *SELECT*, *CALCULATE*, *RESELECT*, and *ASELECT* commands were used to:

[3] Canadian legs of shipment routes were not processed.

- Select the arcs belonging to the current owner, **or** if the current owner was associated with a group of owners that share tracks, then select the arcs belonging to the whole group. (Only the six most important routing partnerships were used). Set the IMPEDE field of each selected arc to its arc length.
- Select the arcs of all the other owners. Set the impedance value of each selected arc to twice the length of the arc.
- Select all unknown owners' arcs. Set the impedance value of each selected arc to three times the length of the arc.

The impedance values were set so that the routing algorithm would first choose railroad tracks of the current owner or group of owners, then choose tracks from the other owners, and finally, choose tracks of unknown ownership. Therefore, abandoned tracks were the least likely to be used.

Routing AML

The routing AML macro created shipment routes by using Arc/Info's Network commands via the ArcPlot module. The following commands were used:

NETCOVER	- to specify the PAREDRAIL network file to be used by the Network commands to create and store the route system tables,
IMPEDANCE	- to specify the IMPEDE field to be used by the Network commands for network impedance values,
PATH	- to find the minimum path between the *from_node* and the *to_node* for each leg of a shipment.

The AML also contained ArcPlot drawing commands (*MAPEXTENT*, *ARCLINES*, *ROUTELINES*) for visually checking the route systems as they were created. Since the drawing time slowed the processing time, only the first few routes were verified for each batch, then the drawing commands were turned off until deemed necessary again.

GENERATE OUTPUT

After calling the routing AML, the main AML used the Tables *SELECT* command with the AML *SHOW* function to check if the route had indeed been created. If so, the route attribute table (RAT) and associated section (SEC) files were *EXPORT*ed via the *INFO* option[4]. If not, a message was written to an error file.

The *FREQUENCY* command was then used to create a non-duplicate list of all the route identification numbers (ROUTEID values) of arcs that the shipment leg had traveled across. The Tables *SELECT*, *RESELECT*, and *UNLOAD* commands were used to write out all nonzero ROUTEIDs with their associated UNIQUE number into a text file in columnar format. The *DROPFEATURES* command was then used to delete the RAT and SEC files because of the

[4] It was later discovered that the exported route systems were viewable in ArcPlot, but not ArcView, since they were no longer attached to the original railroad network file.

Arc/Info limit on the number of Info files[5]. Therefore, an output text file was created for every leg of a shipment that contained nonzero ROUTEIDs.

Multiple error files and status reports were also created while processing the shipment records. Information from these files was used to re-process shipment legs when possible.

After all the records were processed, UNIX 'cat' commands were used to concatenate all of the ROUTEID output files into one large file. It was *LOAD*ed back into Tables and *SORT*ed by UNIQUE number, and *UNLOAD*ed again to a text file and shipped to the customer.

CONCLUSION

The initial objective of this phase of the railroad capacity project was to develop a GIS application to simulate routing railroad shipments and produce a list of specialized route identification numbers for the customer in less than two months. The GIS Team accepted the proposed project with the mutual understanding that this was a high risk research and development project. (It was not known at the onset if the desired product was feasible.) However, the initial GIS application was developed in less than two months, and the project would have been completed on schedule had it not been for the large amounts of time necessary to process the data. In spite of this, the end product was achieved and the process also pioneered the development of other beneficial products.

The following output files were created for Phase 2 of the railroad capacity project:

1). Text file containing attribute information for arcs considered important for calculating the line-haul capacity of selected railways:

FIELD NAME	*DESCRIPTION*
UNIQUE	Unique Shipment Identifier Assigned by Dr. Mark Burton from the customer shipment records
ROUTEID	Dr. Mark Burton's Route Identification Number from the railroad network arcs

2). Exported Arc/Info route system files
3). Error text files
4). Status reports

Other products created for this project were:

1). 1:100,000 scale railroad network with specialized attributes in Arc/Info format
2). Railroad stations in Arc/Info format
3). AML software for routing railroad shipments
4). Color plots of routes deemed to be within the top 100 rail capacity indicators

In conclusion, similar future projects should be given ample time and funding for developing and implementing more efficient routing methods, as well as consulting experts in the GIS Transportation business.

---

[5] It was later discovered that this could be avoided by using the *NETCOVER* and *PATH* commands differently.