

Title: Spatial and Temporal Analysis Of Transportation Statistics (STATS)

Paper No: 1591

**Jacob-Satin Jacobs
&
Julie Henry**

Abstract:

Los Angeles Metropolitan Transportation Authority (Metro) required the development of a Geographical Information System that would spatially and temporally track the supply (scheduled buses) and the demand (ridership) for their buses on all their bus routes. The analysis could be for any given time period and results had to be at the city block level. The completed system would assist Metro in adjusting the number of buses and coordinating schedules in order to anticipate demand. Metro partnered with Integrated Spatial Solutions Incorporated (ISSI) to develop the Spatial and Temporal Analysis of Transportation Statistics (STATS) GIS system. This paper will outline the methodology used by ISSI to create STATS and will give examples of its use by Metro.

Body:

The Los Angeles County Metropolitan Transportation Authority (Metro), serving the urbanized area of Los Angeles County and parts of adjacent counties, is the region's largest transit agency. Metro services 1.3 million daily unlinked passenger boarding, including both bus and rail service. Metro's Service Performance Analysis Department is tasked with collecting performance data and assembling meaningful information and performance indicators to assist management in directing bus operations. Metro maintains two sets of data that are utilized by the STATS application, HASTUS and Automatic Passenger Counter (APC). The HASTUS database stores scheduling data and the APC database stores ridership information. Metro sought the services of a GIS consulting firm to automate the process of loading new data at each schedule change, incorporating demand data from the passenger counting system, and developing an application to visualize ridership and schedule data. ISSI designed, developed, and implemented Spatial and Temporal Analysis of Transportation Statistics (STATS) GIS to meet Metro's requirements.

The HASTUS system uses Thomas Brothers street network as its base data and Metro's transit system for creation of bus route timetables and operation assignments. Bus schedules change every six months. Each six-month segment is referred to as a booking period. For each new booking period, data is exported from the HASTUS system in the form of comma delimited ASCII files. The exported HASTUS

files are imported into a server running AIX 5.3, Oracle 10.1.0.4 64 bit, and ArcSDE 9.1. ISSI created a set of procedures and scripts to automate the importing process. Once the import process is complete, geographic features must be created for the scheduled service.

Metro's scheduling data is based on the Thomas Brothers street network. However, the spatial data within HASTUS does not correspond directly to the Thomas Brothers street network. Due to the spatial discrepancy, there is no way to identify and link features in the street line feature class to the corresponding Metro data. The solution to this problem is to create a feature class from the virtual time points for each pattern existing in the HASTUS database.

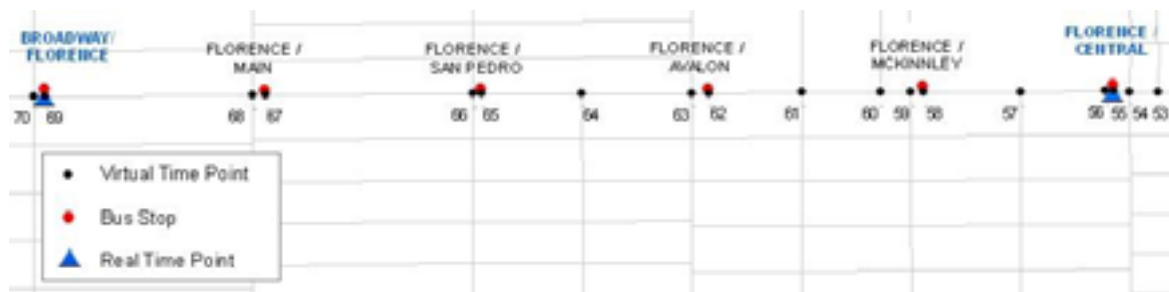
The Virtual Time Point table is the primary source for creating geographic features. This table contains latitude and longitude that defines the sequencing of time points and bus stops for a specific line, route, pattern, and sequence. A line indicates travel from one area to another along a general corridor and is made up of a collection of routes, patterns and segments. A route is the specific path along the line and is comprised of a group of patterns and covers the same stops at least once. A pattern is a variation in a route along the line and covers one or more routes in a line. The pattern is also a spatial representation of the exact space traversed for a given trip. A pattern is made up of numerous segments. A segment represents vehicle operating times between two real time points on a pattern for a given line. A real time point is the starting bus stop or ending bus stop for a segment. The majority of real time points are indicated on Metro's published bus schedule. A bus stop is the geographic location along a segment where passengers may board the vehicle.

There are four phases to creating geographic features. The first phase involves creating customized scripts that are used to generate geographic features from the Virtual Time Point table. These geographic features are then related back to the HASTUS non-spatial tables. The Virtual Time Point table is queried for a unique combination of line, route, and pattern number. Each record contains a "from time point", "to time point", and "sequence number", along with the corresponding latitude and longitude. The Virtual Time Point table defines the sequencing of real time points and bus stops and the direction. The first step in creating geographic features is to generate a line feature for each pattern. Using Visual Basic scripting language, a feature class is created for each unique line. Each line feature class contains a record for each pattern within that line. Lines, routes, and patterns can overlap one another. Therefore, measurement values must be added to each line feature to facilitate the use of dynamic segmentation in future processing.

The figure below depicts virtual time points and their sequence numbers along a portion of line 111, route 111, and pattern 15. There is a virtual time point located at the Florence/McKinnley and Florence/Central bus stops and the Florence/Central real time point.

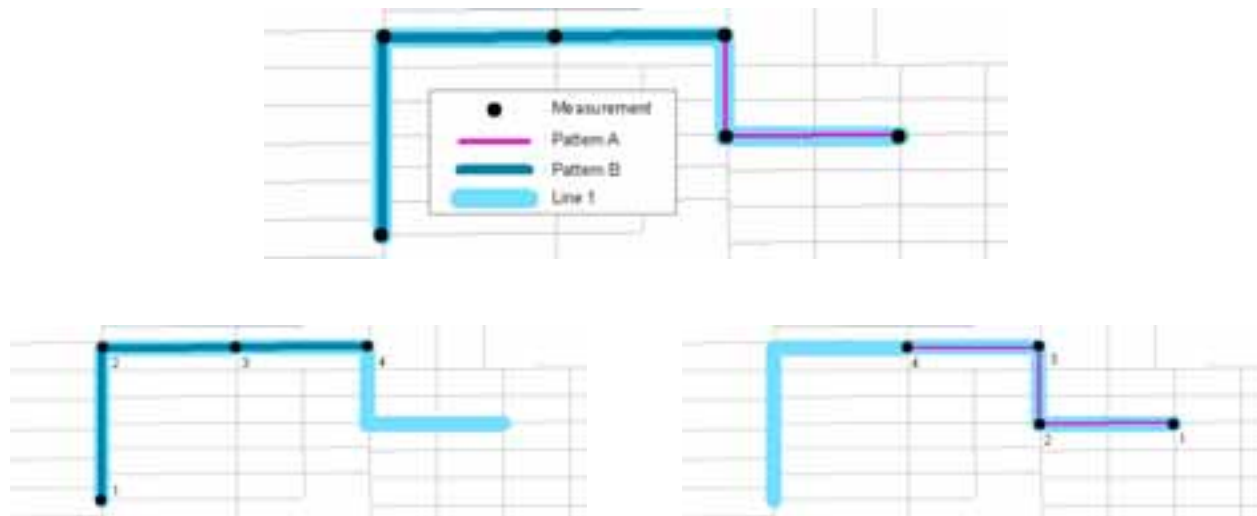


The image below depicts a segment of line 111, route 111, and pattern 15. The segment runs between the Broadway/Florence and Florence/Central real time points.



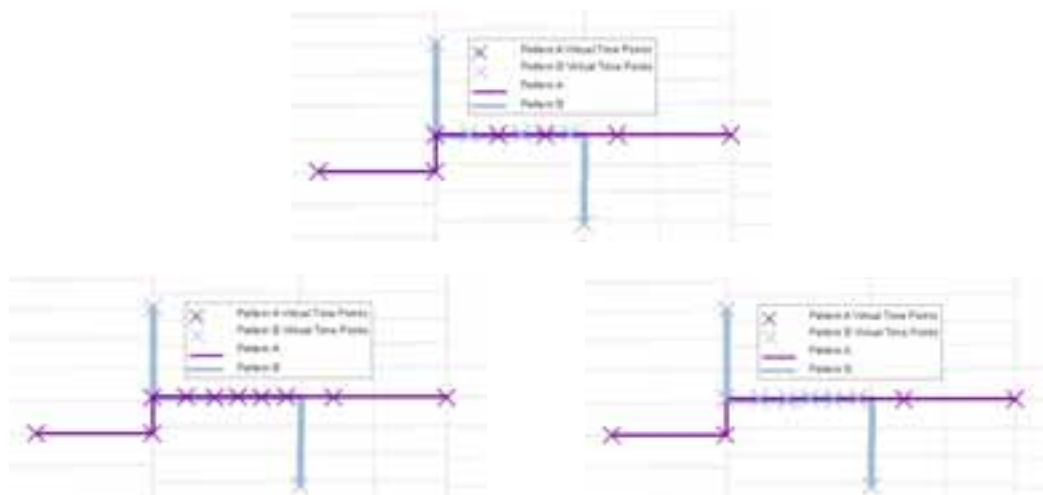
The second phase in creating geographic features requires assigning measurement values to allow for dynamic segmentation in future analysis. In an effort to ensure that all pattern measurements are uniform, measures are assigned using coordinate priority from the lower left corner of the active view. However, due to the nature of Metro's pattern geometry, assigning measurement values based on the lower left corners is not perfect. At this point, the patterns need to be checked manually to ensure that they are running in the correct direction. If all patterns for a particular line do not have their measurements traveling in the same direction, they must be manually switched.

The diagram below shows an example of measurements assigned incorrectly due to the nature of the geometry.



The third phase of geographic feature creation requires assigning virtual time points and measurements to all lines and patterns that occupy the same geographic space. All virtual time points are plotted using their latitude and longitude. Each pattern is buffered and all virtual time points that fall within that buffer are selected. A table is created from the selected virtual time points and includes their latitude, longitude, and measurement values. This table of virtual time points is then sorted based on the measurement value. Next, a line segment table is created by using each point and the point that immediately follows it for each pattern that consists of a separate record for each segment. Once the segment table is complete the final feature can then be generated.

The figure below is an example of how additional virtual time points and measurements are assigned to each pattern. The additional time points are necessary when two patterns occupy the same geographic space during the same period of time. This ensures accurate results in future analysis.



The fourth phase of the geographic feature creation results in a feature class for each pattern. Using the iFeatureDataConverter2 method, a linear feature is created using the beginning and ending XY value of each record in the segment table. The end result is a feature class that contains a record for each line segment and its corresponding measurement value. Additional fields are then added to the feature class that will be populated during the analysis process. It is possible that zero length features may be created during the processing due to rounding of measurements. These zero length features can cause problems during the analysis process. Therefore, the final step in the fourth phase is to remove records with zero length features.

Metro maintains its ridership data in the APC database. When a bus arrives at a stop and opens its door, the Automatic Vehicle Locator (AVL) system passes the time and location coordinates to APC. The APC system counts passengers getting on board (boarding) and off (alighting) with sensors installed in the passenger doorway of each bus. When the bus doors close, the system stops counting until the next stop. The data is offloaded at the respective operation divisions via the Wide Area Network (WAN) and sent to Metro headquarters for processing.

The information collected from APC is used to provide data for required Federal Reports, such as the National Transit Database, and to provide data used to analyze the actual daily performance of the bus system compared to the scheduled daily operation. The data and reports are also used for short and long term planning of bus routes, Metro Rail lines, traffic engineering, marketing, and cooperative business ventures. Data from APC is vital to the funding, operations, and planning of bus service.

Metro periodically exports post-processed APC data in the form of comma delimited ASCII files. The exported APC files are imported into a Metro server running AIX 5.3, Oracle 10.1.0.4 64 bit, and ArcSDE 9.1. ISSI created a set of procedures and scripts to automate the importing process. Once the import process is complete, APC data can be used with HASTUS data for analysis for the corresponding booking period.

The STATS application incorporates HASTUS and APC data and allows the user to visualize ridership and schedule data. STATS runs on a client PC running ESRI's ArcEditor 9.1. The Metro server running AIX 5.3, Oracle 10.1.0.4 64 bit, and ArcSDE 9.1 serves as the backend for the application and allows multiple users to run STATS simultaneously.

STATS has a Visual Basic form on the front end. When the form is loaded it prompts the user for a user name and password. The users information is used to make an ODBC connection to the Metro server. If the connection cannot be made the form closes. After the connection is made, an SQL query is executed based on the Booking ID to return all unique “Trips” within that booking period based on the Weekday schedule type. A unique trip is each combination of line id, route id, route variant or pattern, direction and weekday type for a revenue type trip. The form contains a series of cascading list boxes for the user to narrow down their selection of route data from millions of records.

The basic query includes booking period, line id, route id, pattern, direction, and day type. There are three schedule day types: Weekday, Saturday and Sunday. Holidays assume the Sunday schedule. The initial query based on the Weekday schedule is then used to determine which service categories to display based on route number.

Service categories consist of: Local (Central Business District (CBD), Local Non-CBD, Limited, Express, Shuttle, and Rapid. They are delineated in the service category table based on ranges of route id numbers. The table contains a field with the service category name, the lower bound of the range and the upper bound of the range. When none of the service categories in the list box are checked, all of the line numbers are listed in the Line ID list. Once a service category is checked each line is then checked to see if it contains a route id within the specified range based on the service category table.

At this point there should only be Line IDs listed in the Line ID box if their associated Route Ids fall within the service categories that are selected in the Service Type box. Local CBD includes routes numbered 0 thru 99 and Limited includes routes 300-399. Only lines containing those route numbers are shown and only those routes are shown. If we were to unclick the Local CBD the Route 2-2 would disappear from the Route Id List.

When Line IDs are checked, the TRIPS table is queried again. If the selected Line ID's associated Route ID falls within a selected service category, then it will be listed in the Route ID List.

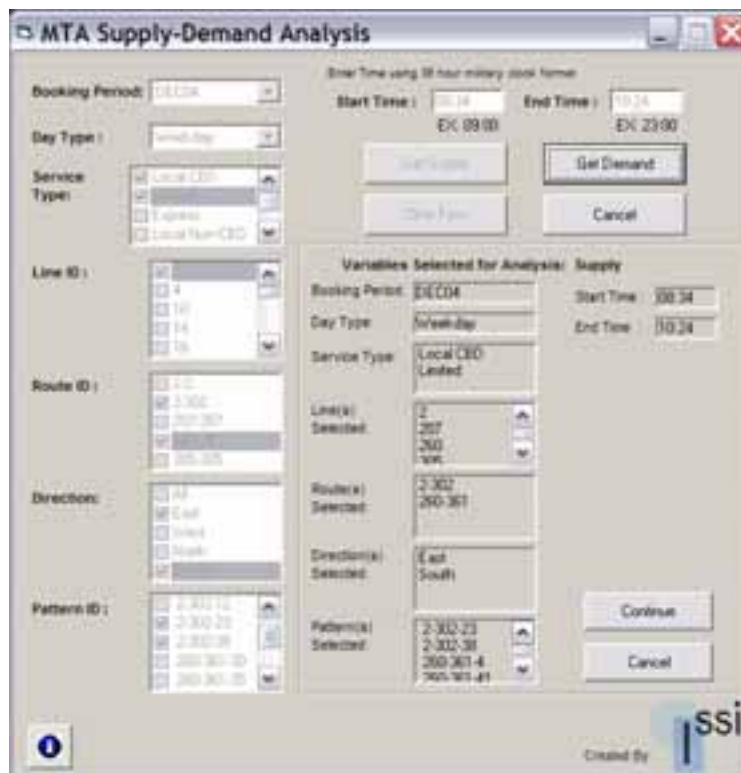
When the Line ID – Route ID is selected from the Route ID box the database is queried for direction because a route can travel in different directions.

The image shows two side-by-side screenshots of a software interface. The left screenshot shows the 'Route ID' list with '2-302' selected and the 'Directions' list with 'All' selected. The right screenshot shows the 'Route ID' list with '2-302', '207-357', and '260-361' selected, and the 'Directions' list with 'All', 'East', 'West', 'North', and 'South' options.

After any direction or combination of directions, are chosen the patterns will be displayed. Once all the Line related criteria are selected a time period for analysis needs to be entered. Schedules run on a 36-hour clock to account for runs that begin before midnight until after midnight of the next day.

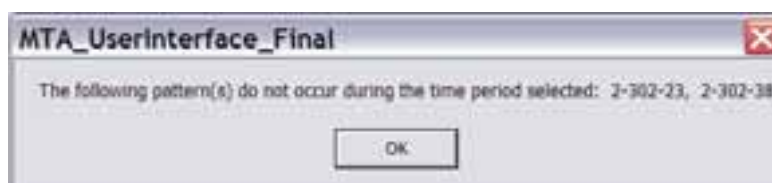
The image is a screenshot of the 'MTA Supply-Demand Analysis' window. It features a 'Booking Period' dropdown set to 'DEC04', a 'Day Type' dropdown set to 'Week-Day', and a 'Service Type' list with 'Local CBD', 'Limited', 'Express', and 'Local Non-CBD' options. The 'Line ID' list shows '2', '4', '10', '14', and '16' with '2' selected. The 'Route ID' list shows '2-2', '2-302', '207-357', '260-361', and '305-305' with '2-302', '207-357', and '260-361' selected. The 'Directions' list shows 'All', 'East', 'West', 'North', and 'South' with 'All' selected. The 'Pattern ID' list shows 'All', '2-302-12', '2-302-23', and '2-302-38' with 'All' selected. At the top right, there are 'Start Time' and 'End Time' fields with 'DEC 09:00' and 'DEC 23:00' respectively, and buttons for 'Get Supply', 'Get Demand', 'Clear Form', and 'Cancel'. The bottom right corner has a logo and the text 'Created By: | ssi'.

Once the time period is entered and the user clicks the desired analysis (Supply or Demand). All of the criteria are displayed to the user for confirmation before running the analysis.



The image shows a software window titled "MTA Supply-Demand Analysis". It contains several input fields and buttons. On the left, there are dropdown menus for "Booking Period" (set to "EXC04"), "Day Type" (set to "Weekday"), "Service Type" (with options "Local CEO", "Express", and "Local Non-CEO"), "Line ID" (with options 4, 10, 14, 16), "Route ID" (with options 2-302, 2-302-23, 2-302-38, 2-302-38-4, 2-302-38-4-1), "Direction" (with options "East", "South", "North", "West"), and "Pattern ID" (with options 2-302-23, 2-302-38, 2-302-38-4, 2-302-38-4-1). On the right, there are "Start Time" and "End Time" fields (both set to "EX 09:00" and "EX 23:00" respectively) and buttons for "Get Supply", "Get Demand", "Clear Form", and "Cancel". Below these, a section titled "Variables Selected for Analysis: Supply" displays the selected criteria: "Booking Period: EXC04", "Day Type: Weekday", "Service Type: Local CEO", "Line(s) Selected: 2, 202, 200, 201", "Route(s) Selected: 2-302, 2-302-23", "Direction(s) Selected: East, South", and "Pattern(s) Selected: 2-302-23, 2-302-38, 2-302-38-4, 2-302-38-4-1". At the bottom right are "Continue" and "Cancel" buttons. The window is created by "ssi".

Once the time period is entered another SQL query is made to determine if the selected route criteria occur within the time frame indicated. The user is informed if no data occurs for particular criteria. This allows them to reselect criteria before running the analysis so the results are not misleading.



The image shows a dialog box titled "MTA_UserInterface_Final". It contains a message: "The following pattern(s) do not occur during the time period selected: 2-302-23, 2-302-38". Below the message is an "OK" button.

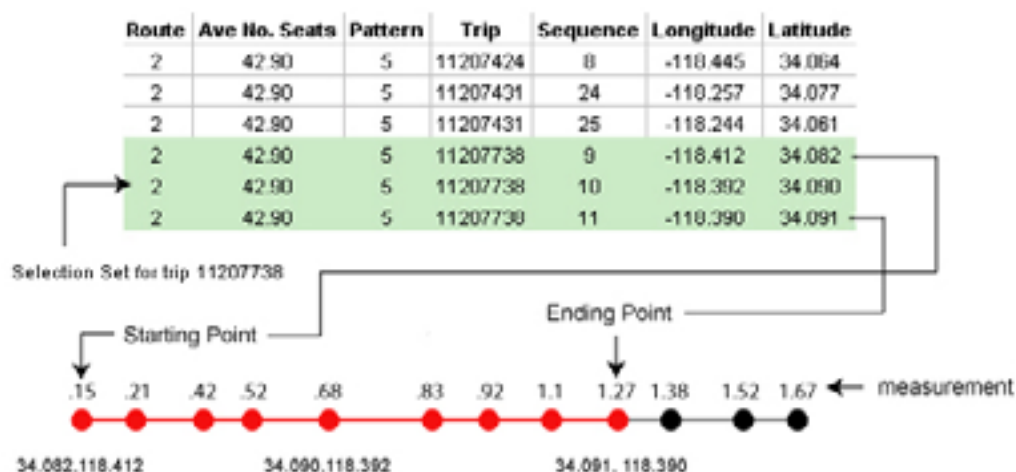
At this point the "Continue" button is clicked to run the analysis based on the criteria displayed in the Variables Selected portion of the form. If the user decides to reselect the analysis criteria they can click "Cancel" to clear the form.

Once the selections have been made on the GUI, the next step is to create complex SQL statements to query the database. The queries access many tables using various primary keys and detailed selection

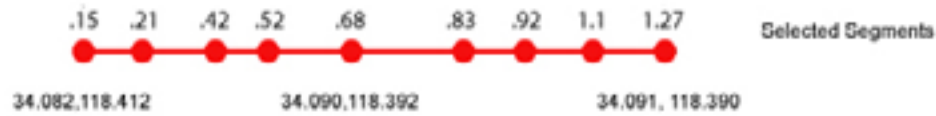
criteria obtained from the GUI. Based on the users selection, whether it is Supply or Demand, there were two sub procedures created. Both of these procedures operate the same but access different tables to extract the necessary data. Each process involves interwoven looping structures. These looping structures involve populating and passing variable arrays between the structures. Once the SQL statements are formed, they are used to create views stored in Oracle, which are later used in the application's processes. These views contain data necessary to perform analysis on the selected Lines, Routes, and Patterns. SDE tables are generated from the SQL views created in the previous step. This is necessary because ArcGIS tables, unlike views, can be queried, sorted, and selected.

The STATS application has multiple users accessing and editing the same feature classes at the same time. Due to multiple users, the next step involves making temporary copies of the selected feature classes. When a STATS user enters their selection, the selected patterns are copied into a feature dataset owned by the user. This ensures the user has permission to edit the pattern feature classes, given that read, write, and query permissions are necessary when performing the analysis.

Once the temporary feature dataset is created, the application enters a looping structure that runs a process for each pattern in the temporary feature dataset. The first step of this process is to create a query that selects all the HASTUS records for each individual pattern selected. This query selects HASTUS records by: pattern type, pattern number, pattern direction, and the start and end time entered in relation to when the pattern is run. A selection set is then created for each individual trip. Each trip's start and end points are extracted using the associated sequence number field. The figure below shows the highlighted records for trip 11207738 sorted by sequence number (9-11). The latitude/longitude values are returned for each record and selected from the corresponding pattern feature class.

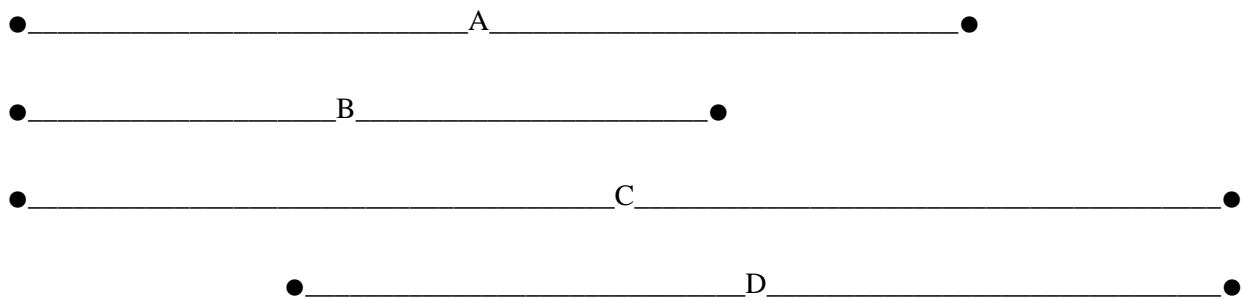


Once the start and end points are determined the corresponding feature segments are selected from the pattern feature class, by using the associated latitude and longitude information. Linear measurements are then acquired from the selected segments.

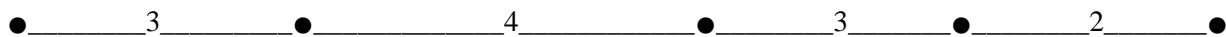


The final step in this looping process is to select each segment that falls between the start and end measurements. A value of one is added to a column that tracks the number of buses that travel on that segment of road. The figure below is a simplified example of combining the trips into one symbolized feature class. This process combines the trips A, B, C and D into the final feature class by adding up the overlapping individual features. When all of the trips are run for each of their associated patterns, the final segments have a total sum bus count.

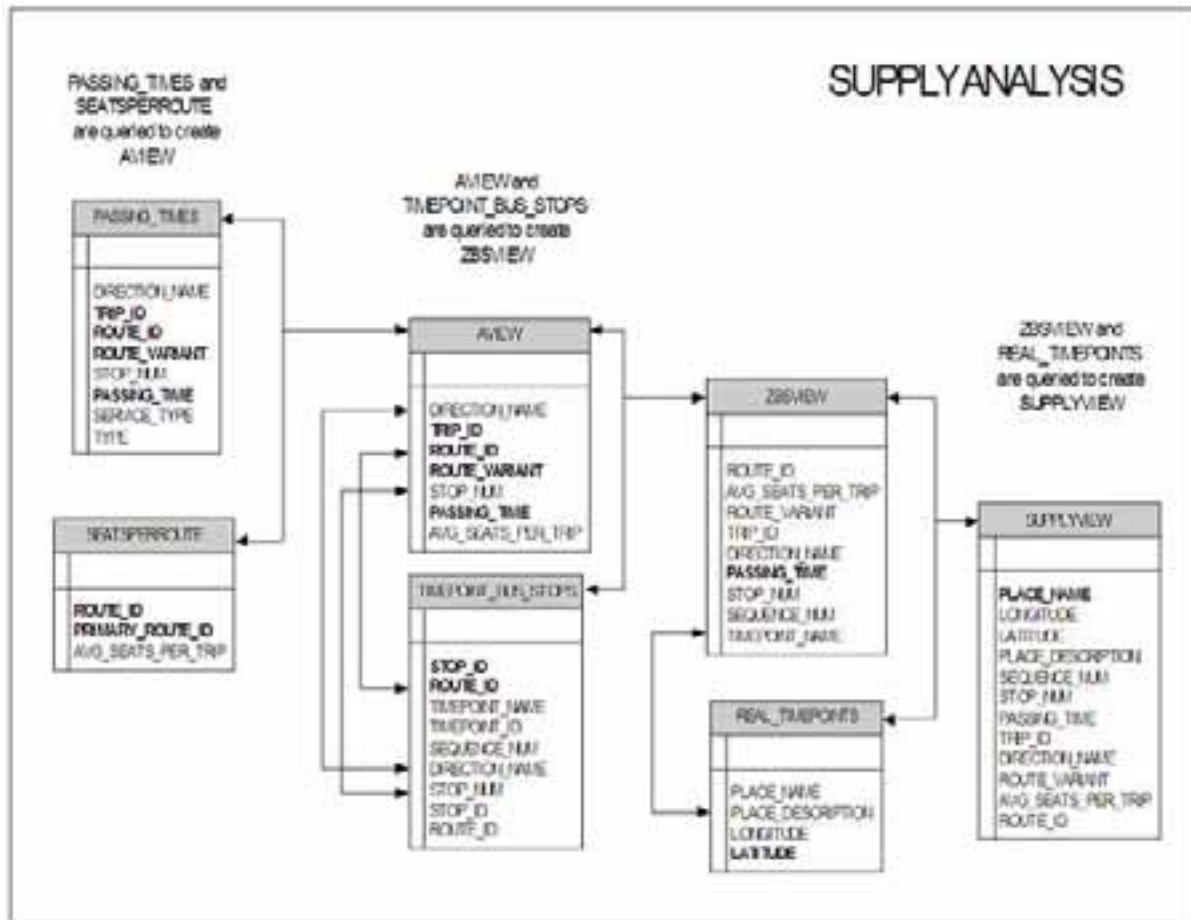
Individual Trips:



Final Feature Class:

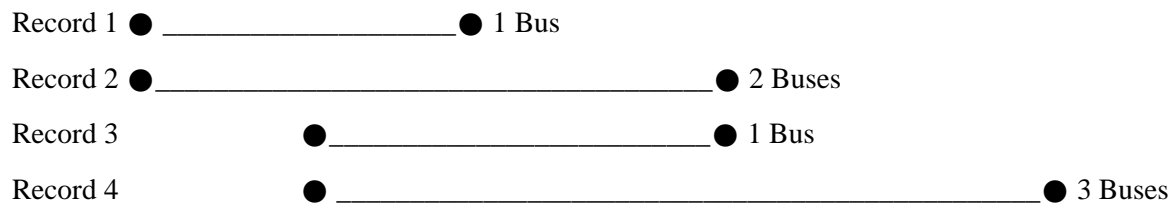


The supply side of the STATS portrays the Total Number of Buses based on the user-input. The user-interface collects the booking period, type of day, line type, line, route id, direction, pattern, start time, and end time. Based on the information provided, a connection is made to the HASTUS database and the following queries are performed:

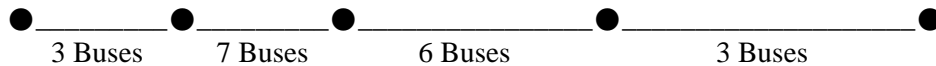


The supply side of STATS counts the total number of buses per pattern. All patterns are merged together and a new feature class is created. The merged feature class is then dissolved based on geometry and the bus count is summed for each line segment.

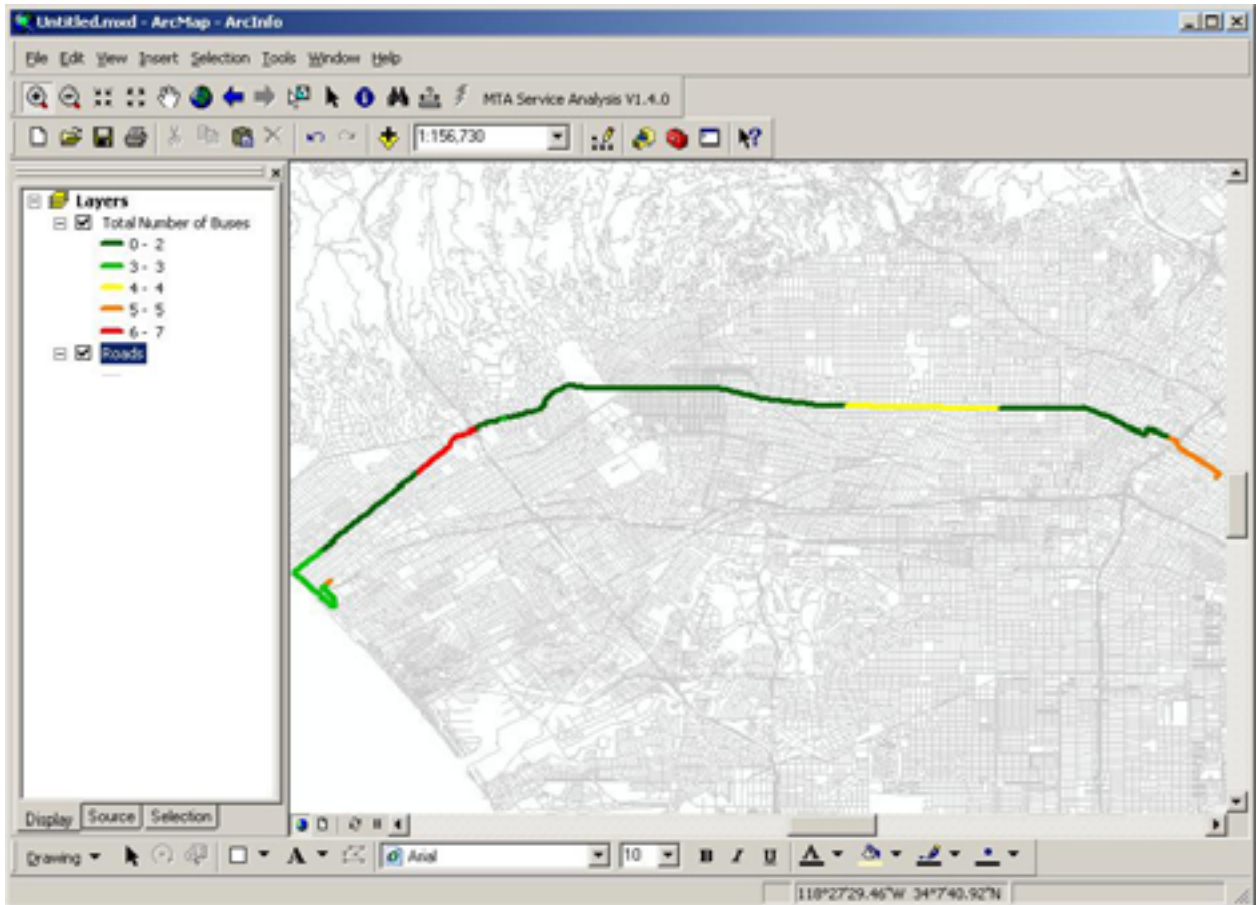
Merged Feature Class:



Dissolved Feature Class:



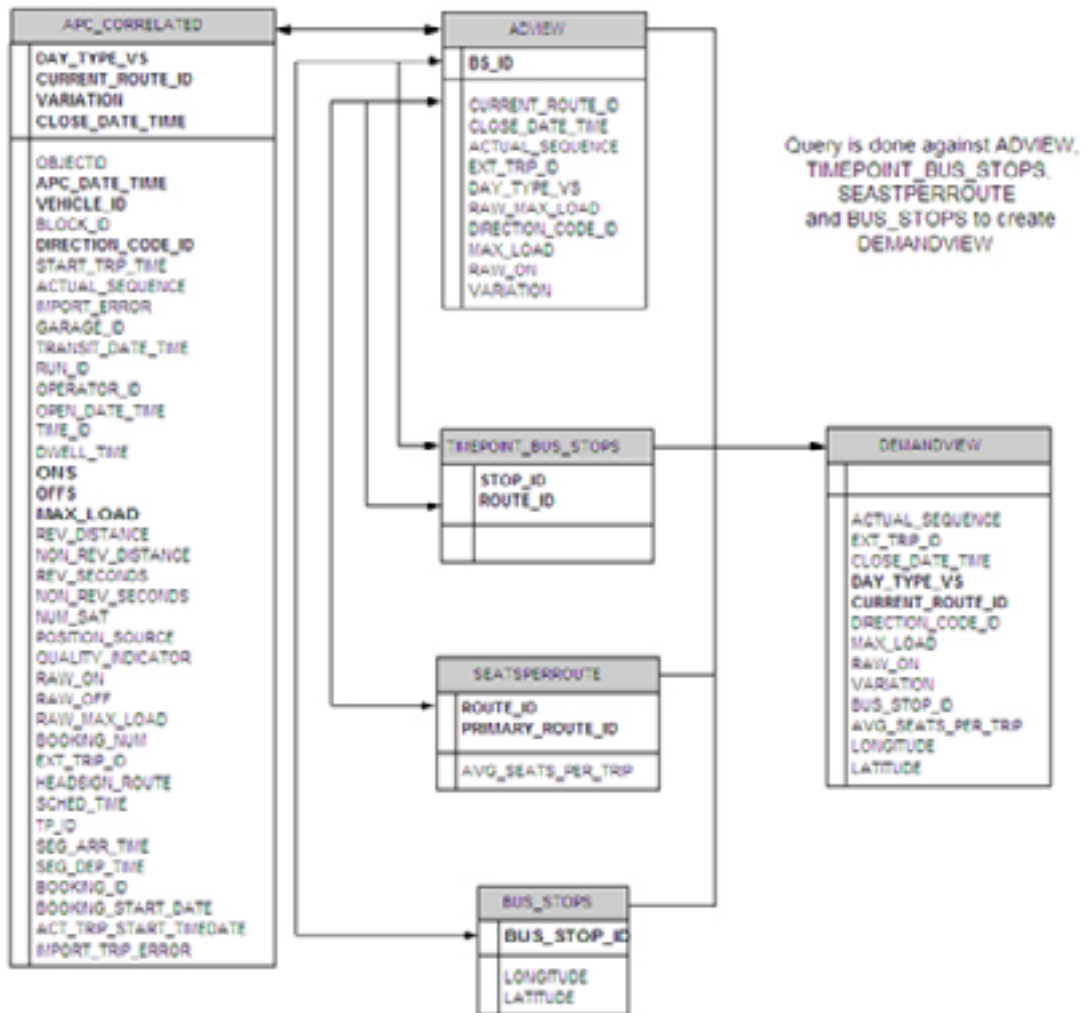
The feature class is loaded into the project and the results are displayed as follows:



The demand side of the STATS portrays the Observed % Capacity Utilized based on the user-input. The user-interface collects the booking period, type of day, line type, line, route id, direction, pattern, start time, and end time. Based on the information provided, a connection is made to the APC database and the following queries are performed:

APC_CORRELATED
is queried against itself
to create
ADVIEW

DEMAND ANALYSIS



The demand side of STATS counts the raw max load per pattern and is displayed as portrays the Observed % Capacity Utilized. The total raw max load is divided by the number of trips per day. All patterns are merged together and a new feature class is created. The merged feature class is then dissolved based on geometry and the average raw max load.

Merged Feature Class:

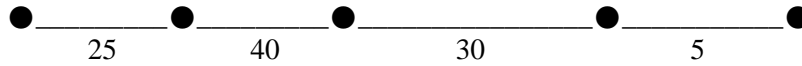
Record 1 ● _____ ● Average Raw Max Load 10

Record 2 ● _____ ● Average Raw Max Load 15

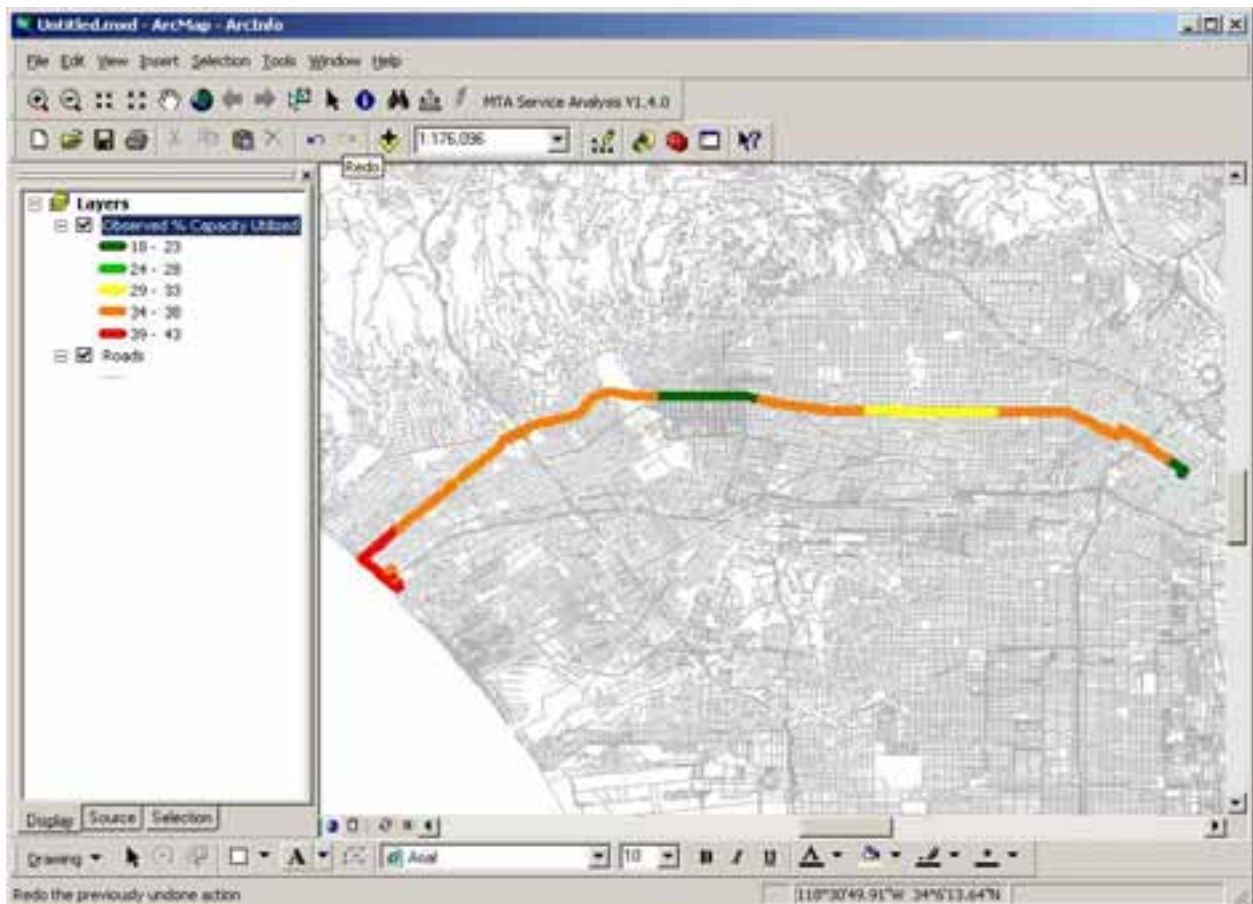
Record 3 ● _____ ● Average Raw Max Load 10

Record 4 ● _____ ● Average Raw Max Load 5

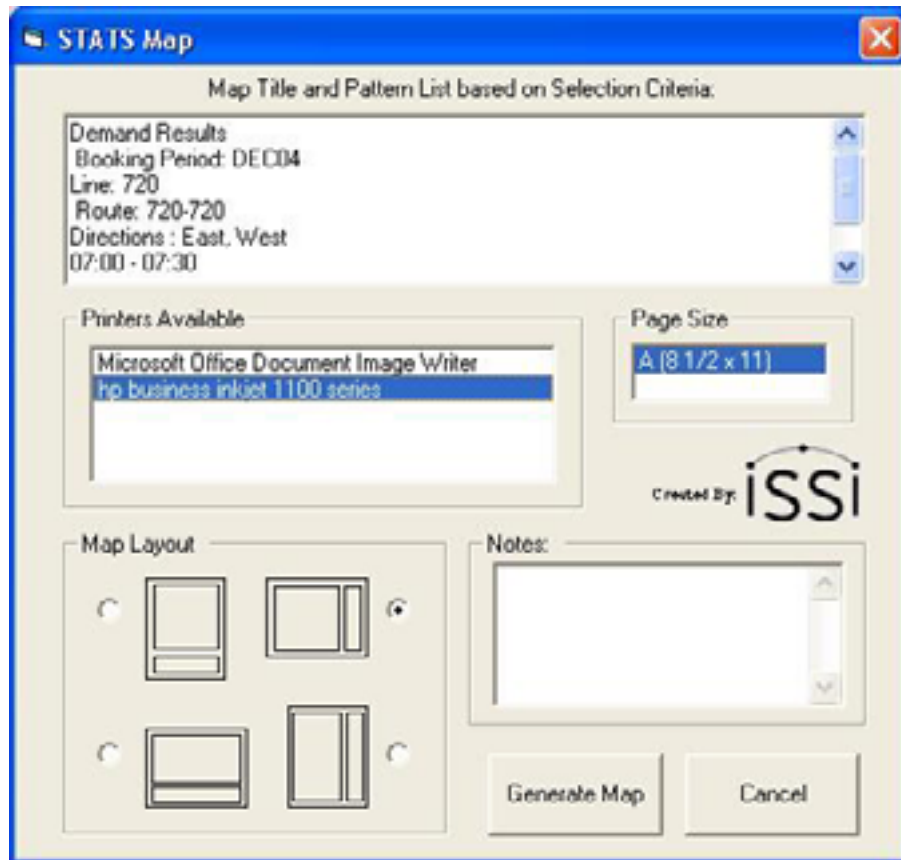
Dissolved Feature Class:



The feature class is loaded into the project and the results are displayed as follows:



After the results of the analysis have been loaded, a message box is displayed and asks the user if they would like to map the results of their analysis. If the user chooses to create a map, a second dialog will appear.



This dialog will search the users local computer for available printers and allow them to select one. Depending on the paper size capability of the selected printer, a list of available page sizes will appear. The user can select from page size 8 ½ x 11 or 11 x 17, orientation portrait or landscape. The following map components will automatically be filled in by the map template: scalebar, north arrow, Thomas Brothers logo and disclaimer, MTA logo, notes field, legend, and title. The title is automatically populated with the user input based on the input from the STATS interface.

Below is an example of the final map results.

