An Introduction to TRANSIMS: *Core Components and New Developments*

Michael Hope

Computational Transportation Engineer, Transportation Research and Analysis Computing Center

Energy Systems Division

Argonne National Laboratory

transims@anl.gov

http://www.tracc.anl.gov/

TRACC - A National User Facility to Meet USDOT Advanced Computation Needs

- USDOT and USDOE transportation research programs, private industry, and state and regional transportation agencies are moving to simulation-based design and analysis for improvements in efficiency, economics, and safety
- Higher fidelity analysis in areas such as crashworthiness, aerodynamics, combustion, thermal management, weather modeling, and traffic simulation require access to state-of-the-art computational and visualization facilities
- Argonne expertise in high-performance computing and transportation system analysis provides the basis for a national HPC user facility and a focal point for computational research for transportation applications

TRACC - High Performance Computing for Transportation Research and Applied Technology

Thermal Management *

Crashworthiness *

Bridge Hydraulics*

Occupant Safety Assessment and Crash Biomechanics *

Bridge Structural Analysis *

Aerodynamics *

Intelligent Transportation Systems*

Road Weather Research *

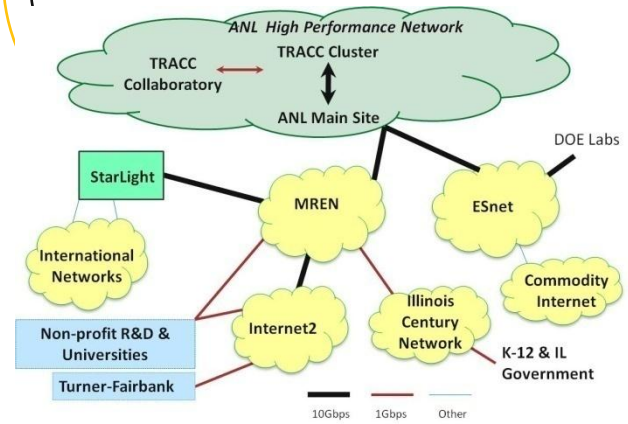
Traffic Modeling *

Transportation Data Archive *

TRACC Is a National USDOT Supercomputing Facility

TRACC High Performance Compute Cluster

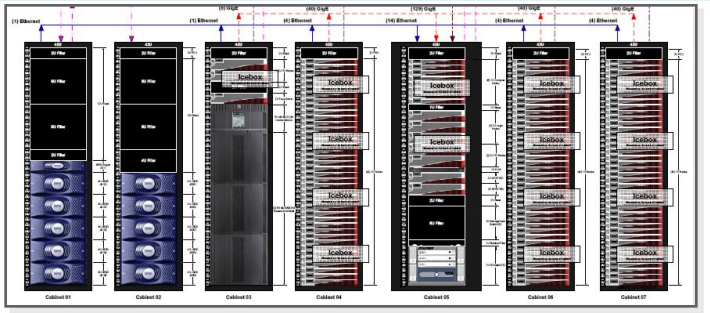
- *1024 core /128 compute nodes*
- 180TB Global Parallel File System Disk Storage
- 160TB Archive/Backup Tape Storage



High-bandwidth connectivity is provided via the Argonne high-performance network to world-wide research and education networks (Internet2 and ESnet)

TRACC Collaboratory - Visualization, Access Grid, and Digital Conferencing

TRANSIMS User Support and Training



- TRACC is providing training courses on TRANSIMS to the transportation research community in the US
  - Training courses are offered approximately 4 times per year in varying locations
  - Participation is free, and training courses are broadcast over the Internet to reach additional users
- TRACC is holding additional training sessions on emerging capabilities through the Internet
- The goal is to build a strong community of expertise

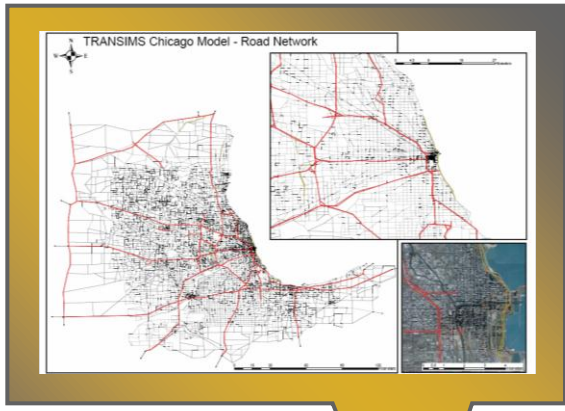
Background

- TRANSIMS is an integrated set of tools to conduct regional transportation system analyses based on a cellular automata microsimulator
- TRANSIMS is an abbreviation for
  - **TR**ansportation **AN**alysis and **SIM**ulation **S**ystem
- The TRANSIMS approach is based on the new paradigm of modeling individual travelers and their multi-modal transportation based on synthetic populations and their activities
- Compared to traditional traffic planning approaches, TRANSIMS requires a significant amount of data and computing resources
- The software was initially developed at Los Alamos National Laboratory; it is now being made available and furthermore developed as an open source project:

  http://www.transims-opensource.net

- Latest Release: Version 4.07, released in July 2010

TRANSIMS Background

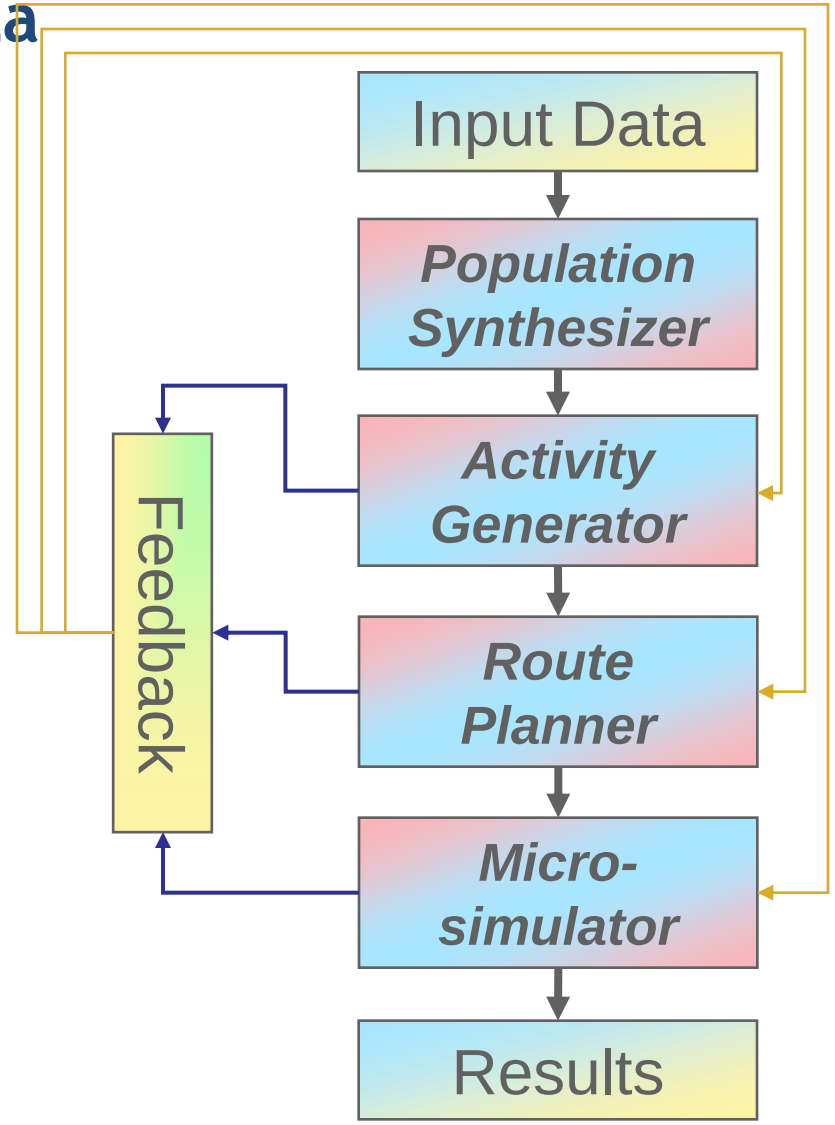


- TRANSIMS is a tool for regional analysis
  - High demands on CPU time and storage capacity
  - Runs on current high end Windows and Linux workstations
  - TRANSIMS accommodates
    - Large road and transit networks (>>100,000* links)
    - Large populations (>>30,000,000* travelers)
- General approach
  - Simulate the travel behavior of each synthetic individual throughout an entire 24 hour period based on representative activities derived from survey data
  - Based on highly detailed road and transit networks, individuals are traced for every second of the day while analyzing their local interactions
  - Routes for travelers are determined by a routing module that considers time-dependent link delays throughout the 24 hour period
  - Microsimulation is used to determine link travel times, rather than volume delay functions

* TRANSIMS has been run successfully with networks and populations of these sizes, but does not enforce actual size limitations. Newer 64-bit operating systems do in fact accommodate even much larger networks and populations, typically limited by memory and operating system constraints.

Scope of TRANSIMS Input Data

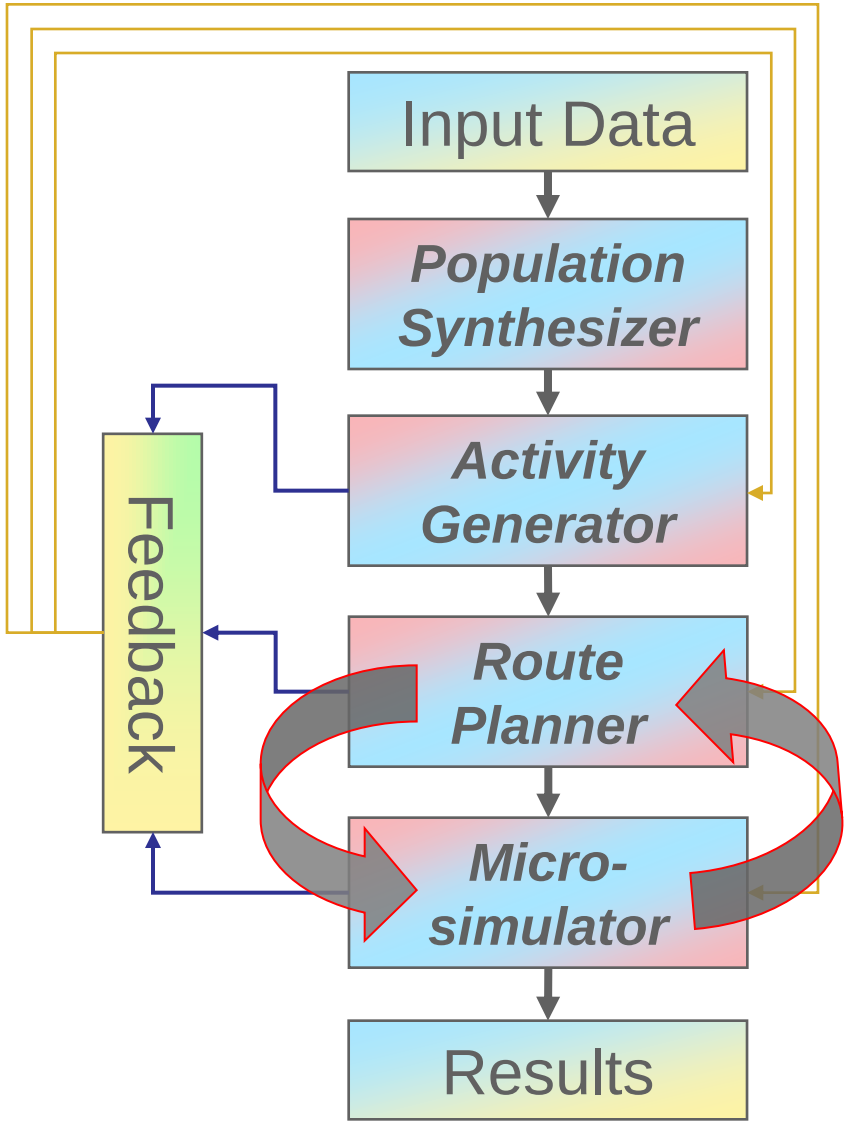
- Input Data for Modules
  - Transportation Network
    - Streets, Intersections, Signals
    - Transit Routes and Schedules
    - Land Use Data, Zoning Information
  - *Transit Lines and Schedules*
  - *Census Data for Population*
  - *Household Activity Surveys*
  - Itinerant Travelers and Trips
  - Vehicle Characteristics and Prototypes
- Current TRANSIMS models are usually derivatives or extensions of existing four step models
  - Populations and activities are bypassed, and trips are extrapolated for each individual in the model



Generalized TRANSIMS Flow Chart

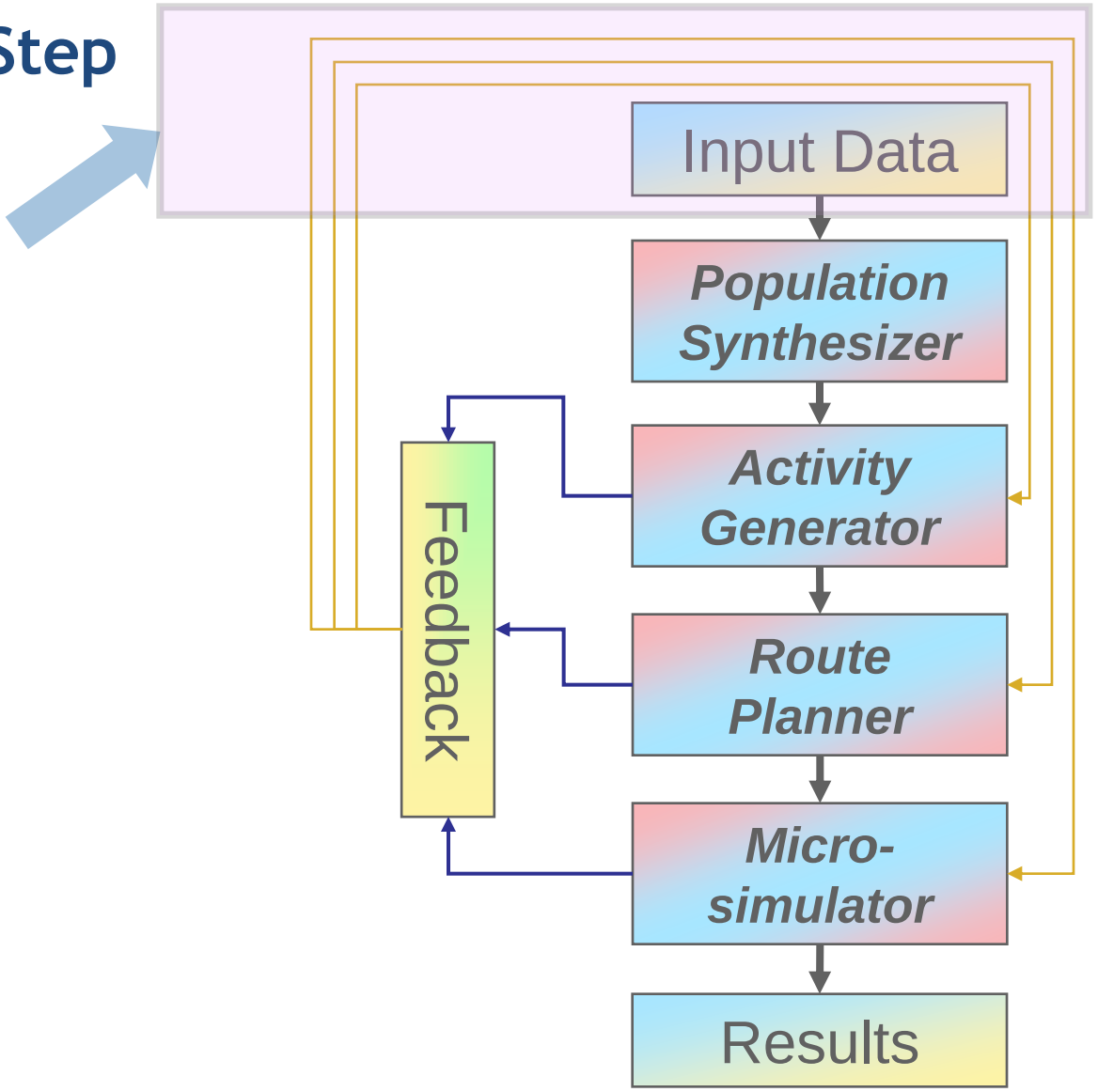
Basic TRANSIMS Methodology

- The goal is to load traffic onto the network and iterating towards equilibrium
  - Travelers cannot achieve significantly better routes when trying to choose a shorter path, meaning that each traveler chooses the route that's best for the overall population
- Important constraint
  - Travelers choose a mode of transportation according to travel surveys; they are not optimizing their travel by choosing modes
- This is simplified
  - Typically, some activities will need to be modified as well to avoid unrealistic travel constraints
- The Activity Generator is typically not used in current models – it needs significant improvements



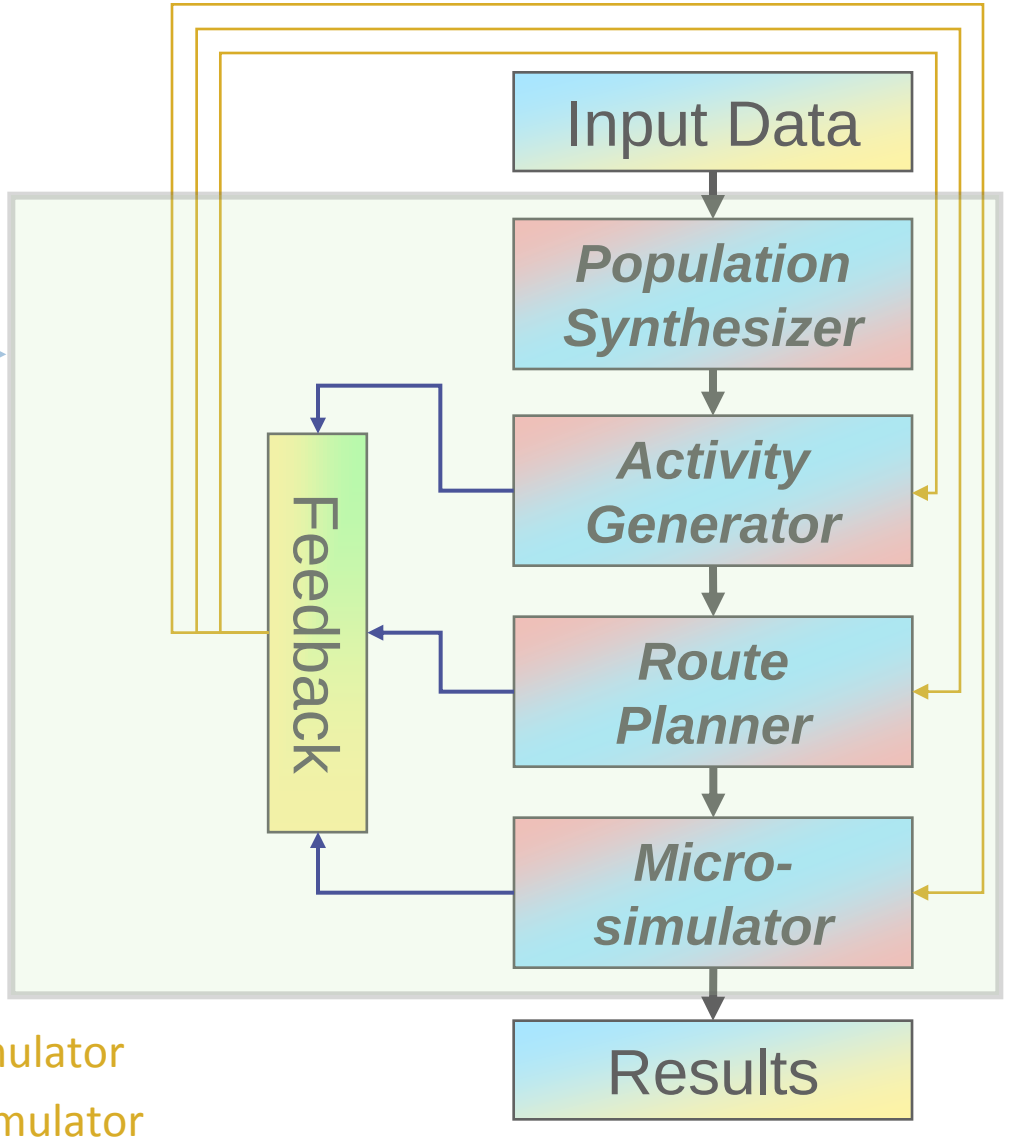
TRANSIMS Step by Step

- Create a Road Network
- Create a Transit Network
- Obtain Transit Schedules



TRANSIMS Step by Step

- *Create a Road Network*
- *Create a Transit Network*
- *Obtain Transit Schedules*
- Methodology #1
  - Obtain Census Data
  - Obtain Activity Survey
  - Run Population Synthesizer
  - Run Activity Generator

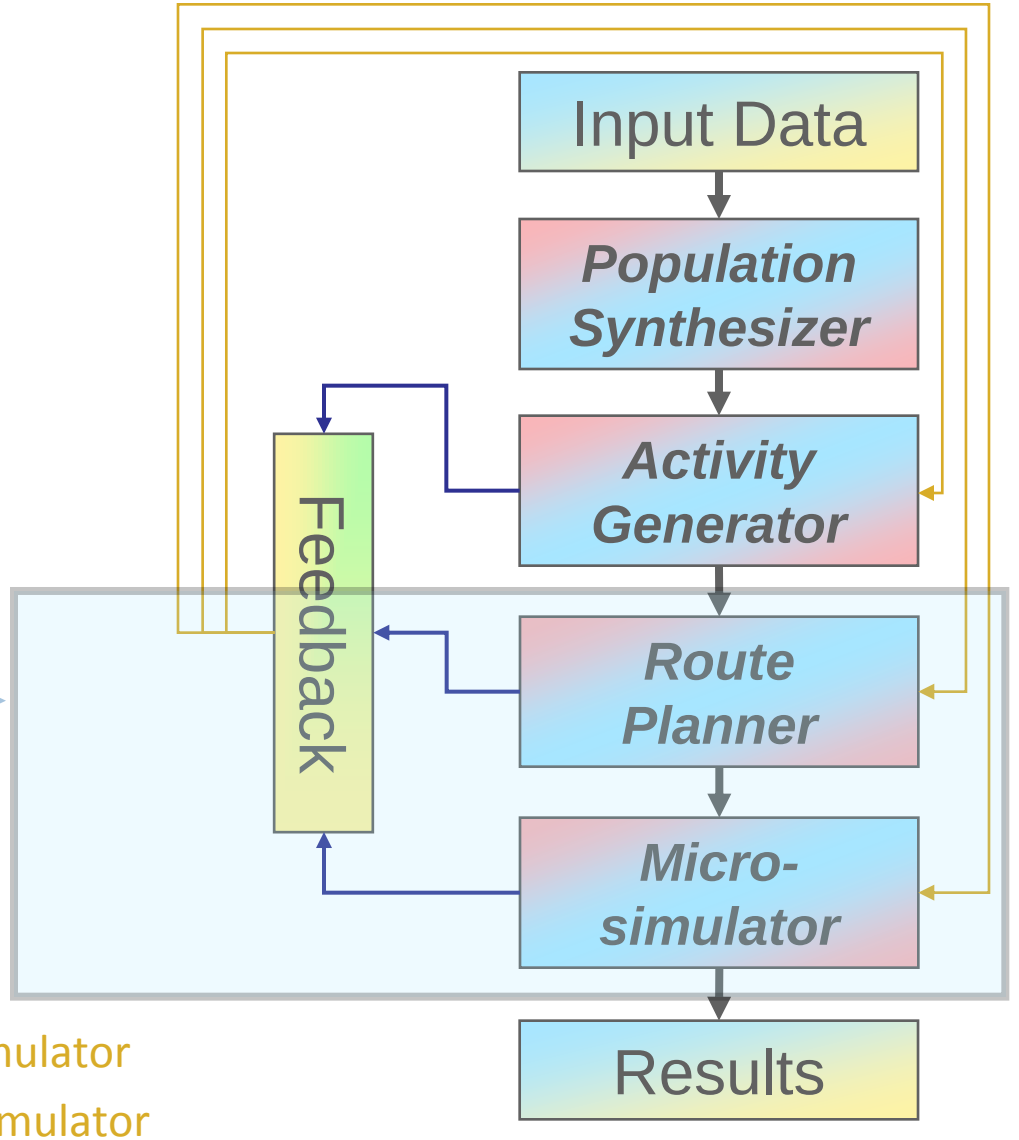


- Create Travel Plans from **Activities** using the Router
- Test the Travel Plans in the Microsimulator
- Iterate Between Router and Microsimulator

TRANSIMS Step by Step

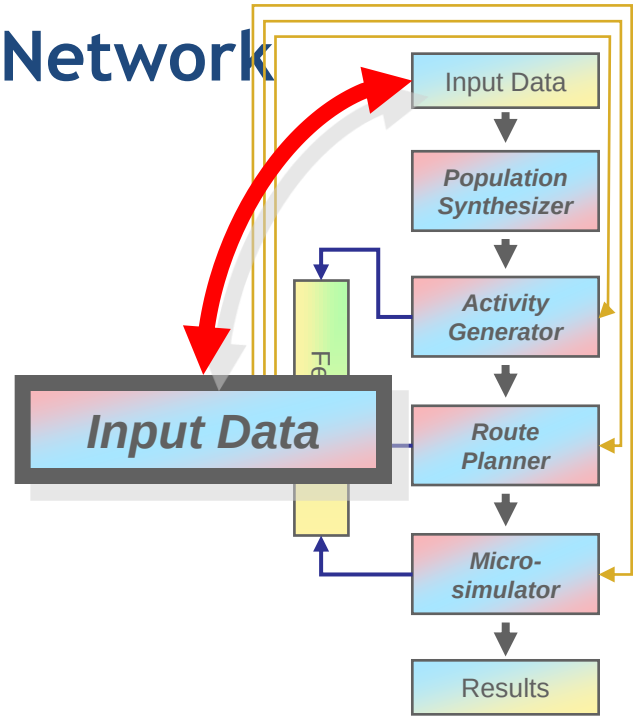
- *Create a Road Network*
- *Create a Transit Network*
- *Obtain Transit Schedules*

- Methodology #2
  - Obtain Trip Tables
  - Obtain Diurnal Distributions
  - Run Trip Converter
- Create Travel Plans from **Trips** using the Router
- Test the Travel Plans in the Microsimulator
- Iterate Between Router and Microsimulator



Input Data – The Road and Transit Network

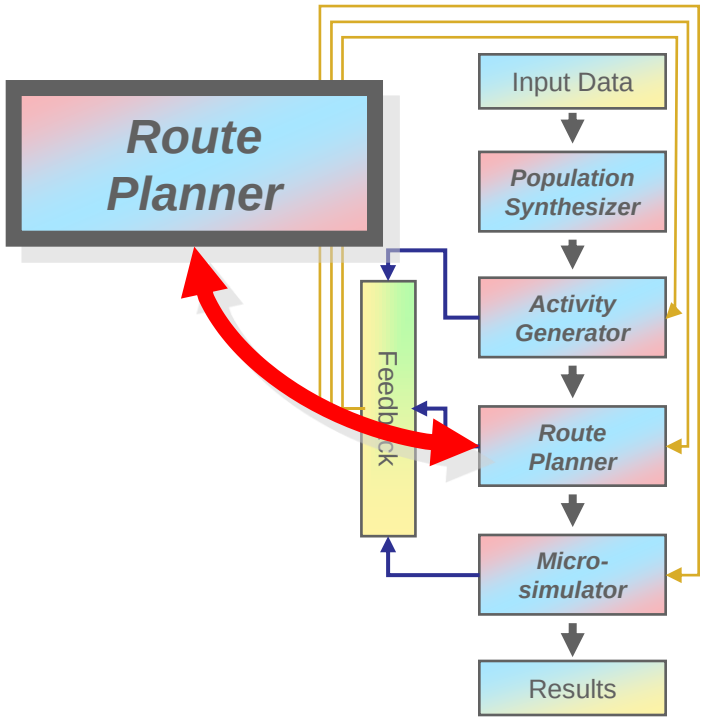
- Typically, street and transit networks are available from metropolitan planning organizations
- Road and transit networks can be exported from other traffic analysis tools into a fairly simple tabular format
  - Nodes, Links, Zones, etc.
- TRANSIMS is able to make use of some common GIS tools and formats with regards to network editing and cutting
- TRANSIMS understands important geographic projection systems (state plane system, universal transverse mercator system, etc.)



- Tools are available to convert TRANSIMS network components into GIS shape files for effective visualization and editing
- Tools exist to single out a smaller area within the simulation and to extract it from the overall network based on complex polygons

The TRANSIMS Route Planner

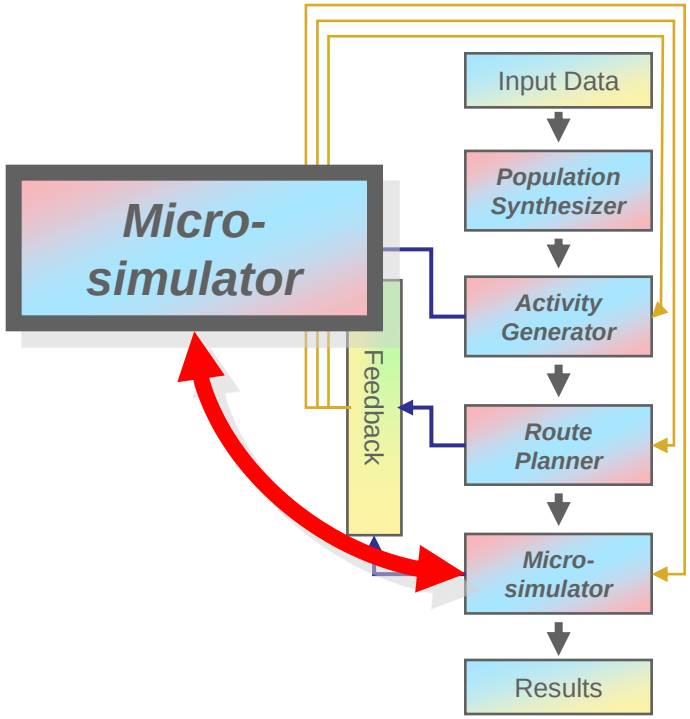
- The TRANSIMS route planner reads the individual activities from the activity generator and determines the fastest route at that time of the day
- Households are routed in a coordinated fashion to allow for ride sharing, such as dropping children off at school.
- The algorithm considers a very detailed time-dependent network based on link delays for each link that vary during the course of the day
- The transportation mode is part of the activity record (or trip record); the router does not choose the transportation mode but finds the best route given the mode of transportation (e.g. choosing the best rail or bus stations for the traveler based on departure details)

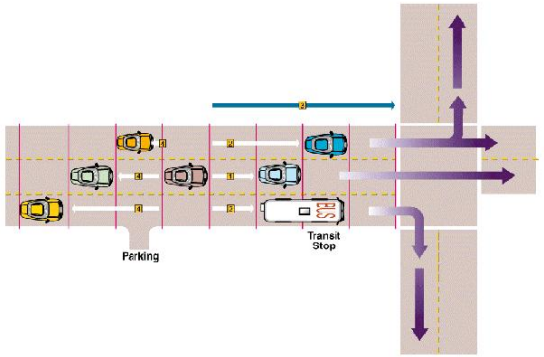


- The route planner is somewhat comparable with services such as MapQuest, but includes time-dependent optimization as well as transit, bicycling, walking, driving, and more

The TRANSIMS Microsimulator

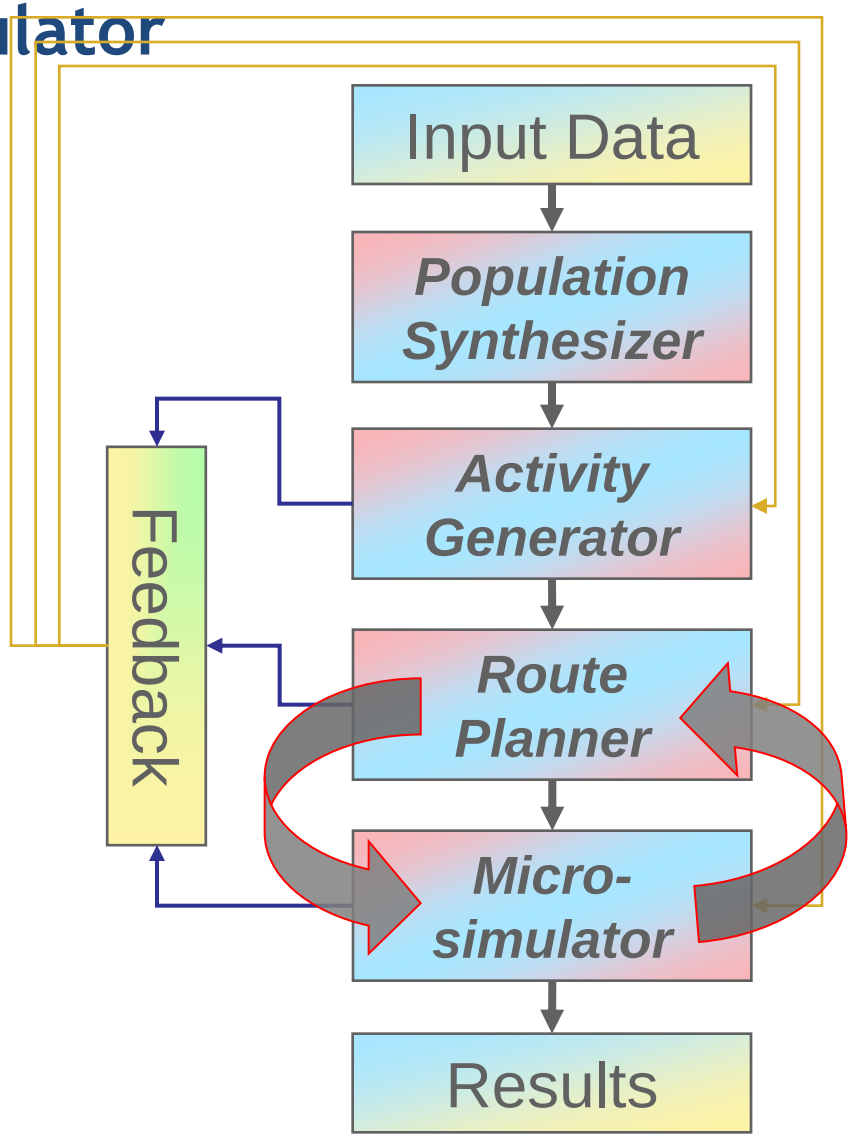
- The Microsimulator executes all travel plans determined by the router on a second by second basis on a highly detailed network
- The Microsimulator uses cellular automata principles to analyze the interaction between individual vehicles, producing emergent behaviors such as traffic congestion
- The Microsimulator can be used to work on the entire metropolitan area or may be used on a smaller region only
  - Additional tools are provided to link the routing process for the metropolitan area properly with the microsimulation of a smaller region
- The Microsimulator produces individual locations of all travellers and vehicles at all times
- The Microsimulator calculates new link delays that are being used iteratively to selectively reroute some travellers in the equilibration loop





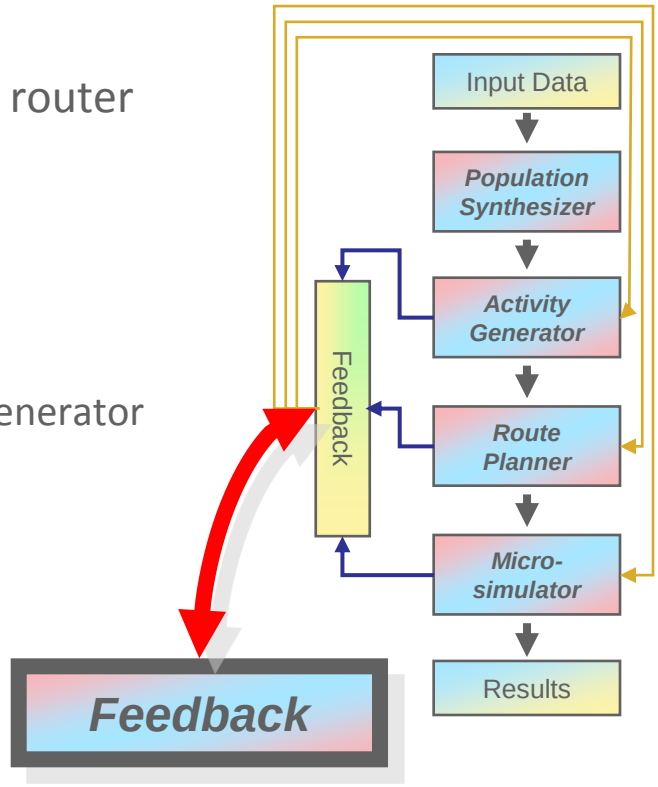
Route Planner and Microsimulator

- The route planner and the microsimulator work in an iterative loop to equilibrate the traffic assignments on the network
- The router starts by using well-known traffic assignment functions (BPR+) to estimate link delays based on the number of trips routed through each link
- The router determines the optimal route for each trip and creates precise travel plans
- The microsimulator tests the interaction between the vehicles while following these travel plans and determines a new set of link delays replacing the ones used previously by the router
- The router and microsimulator iterate until equilibrium is achieved



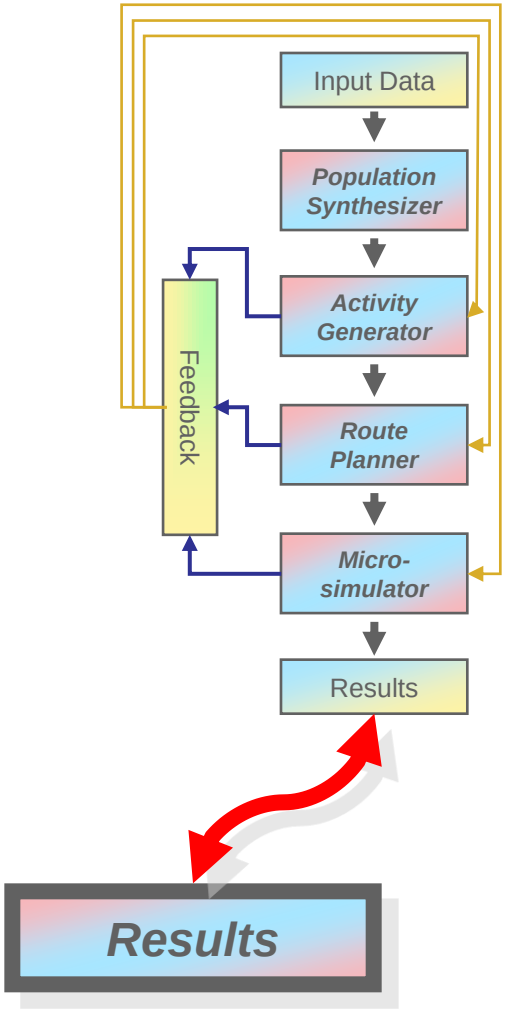
Feedback

- The TRANSIMS equilibration process iterates between router and microsimulator
- Some routes are not feasible, e.g.
  - Significantly longer than dictated by the survey data
  - Not feasible based on the given transportation mode
  - These trips or activities are passed back to the activity generator to determine appropriate alternatives
- In the Microsimulator, vehicles can stall because they are unable to change lanes or make turns
  - Passing the households that own the vehicle back to the router for new routing suggestions may solve the problem
  - Some plans cannot be followed because of time-dependent road closures and other triggers
  - These are the famous “lost cars” – a much criticized flaw
- Tools are available to select households for rerouting based on many criteria



TRANSIMS Results

- TRANSIMS can create aggregate results comparable to traditional tools
- The microsimulation leads also to highly detailed snapshot data, e.g. the exact location of every traveler at any given time
- The amount of data is difficult to comprehend, so effective visualization is essential
  - fourDscape and the Balfour visualizer
  - ArcGIS and similar GIS tools (uDig etc.)
  - The original TRANSIMS visualizer
  - Google Earth and NASA World Wind
  - Tools currently being developed at NCSA
    - Metropolis
    - Ray-tracing
  - Batch movie tools developed at Argonne
  - NEXTA, developed by the University of Utah
- New tools developed at TRACC as part of TRANSIMS Studio



TRANSIMS Challenges

- TRANSIMS models are significantly more complex than traditional transportation models
  - Data intensive, although much of the detailed network is generated based on rules by default
  - TRANSIMS is a research tool, and needs to be open for future extensions as well as integration with other tools
  - Due to the microsimulation approach replacing the volume delay functions, CPU demand is enormous
  - TRANSIMS requires some basic programming skills compared to other modeling software
  - Technical support is limited due to the open source nature of the project

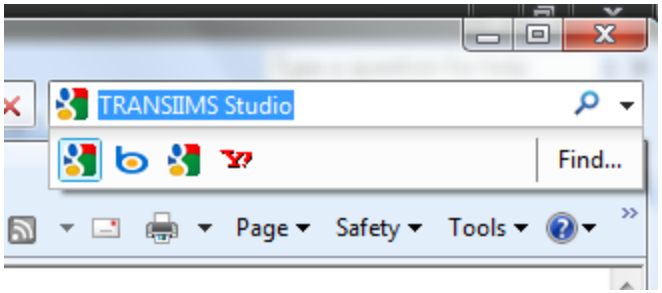
The Original TRANSIMS Software Design

- TRANSIMS consists of a set of over 60 executables that read and write files for each other's consumption
  - This approach makes it easy to extend functionality or interface with other software
  - Each executable is configured using a specific control file, with a total of about 1500 different control keys
  - TRANSIMS by itself provides a rudimentary script and control file writer (RunSetup) that is rarely used
  - While the executables and data files are cross-platform compatible, the user's scripting in batch files or shell scripts is not cross-platform compatible
  - Experience shows that it will take 6 months to a year to become an effective TRANSIMS user

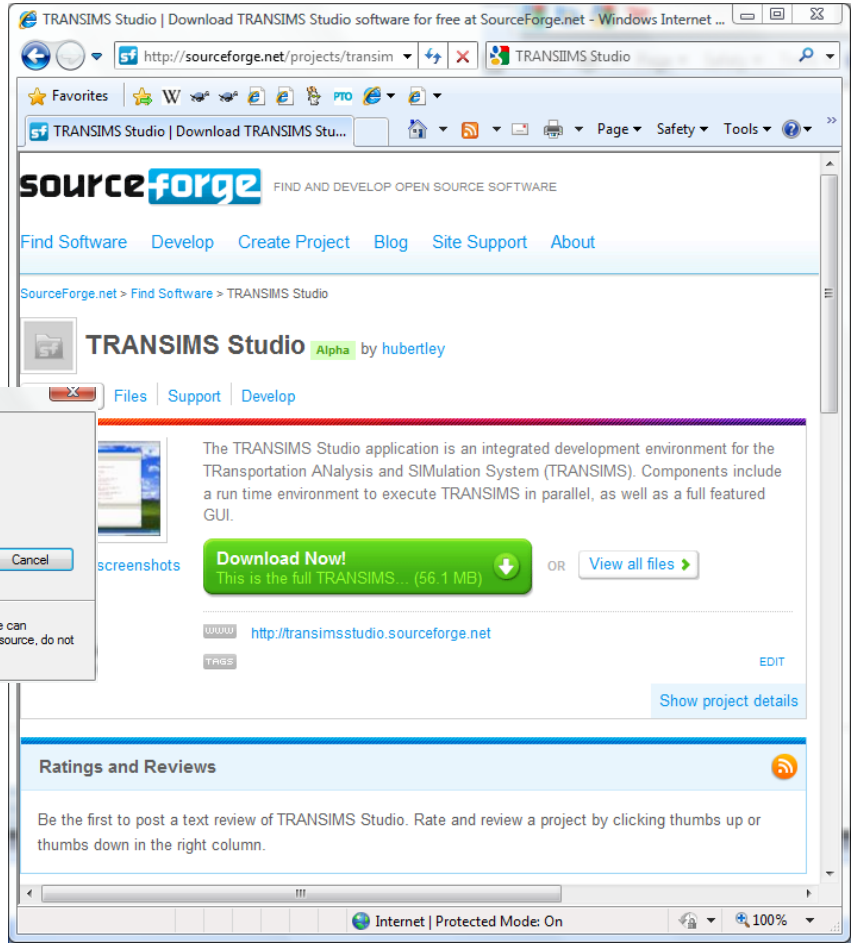
The TRANSIMS Studio Software Application

- TRANSIMS Studio is an add-on to unmodified TRANSIMS executables that simplifies the use of TRANSIMS significantly
- It consists currently of four major components
  - **Transims RTE:** A python run time environment module that provides a reasonably simple API to provide full cross-platform capability for TRANSIMS in an extensible object-oriented framework (not to worry – this sounds complicated but is simple compared to traditional use)
  - **Transims GUI:** A graphical user interface that allows access to all source and generated data files and controls the program flow for a complex TRANSIMS model
  - **Transims VIS:** An interactive visualizer that allows also for the generation of videos through a graphical user interface
  - **Transims EDT:** An integrated network editor that is fully aware of the TRANSIMS network topology

How to get TRANSIMS Studio



- Google for “TRANSIMS Studio”
- Download the installer from SourceForge
- Administrative install needed
- Optionally, download the Alexandria dataset as an example

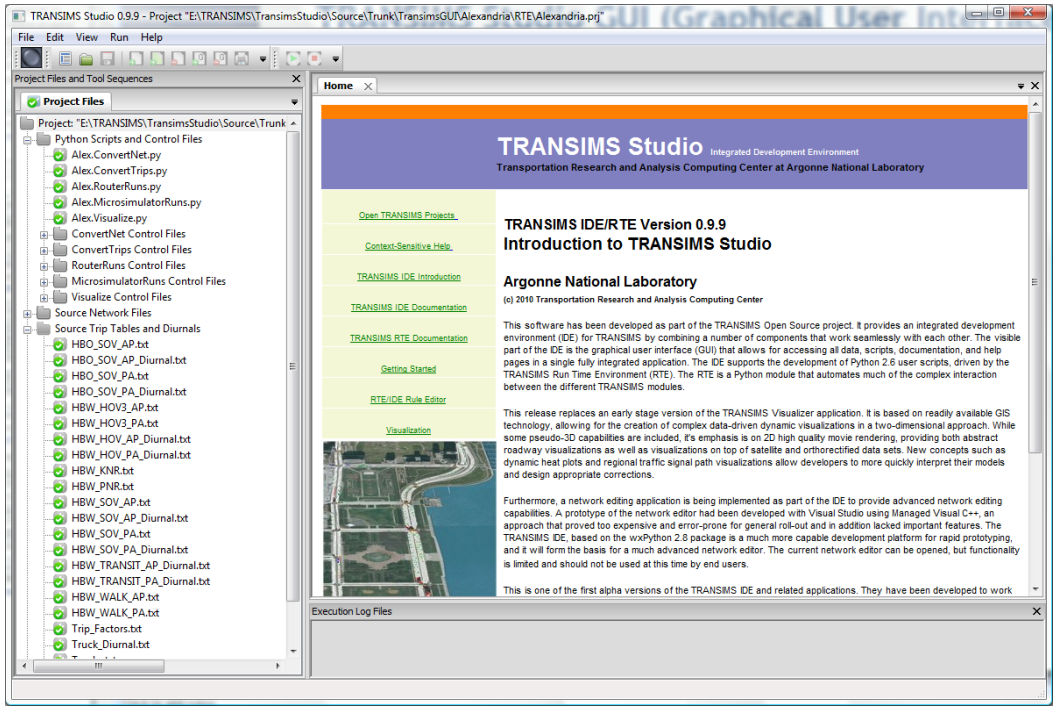


TRANSIMS RTE (the Run Time Environment)

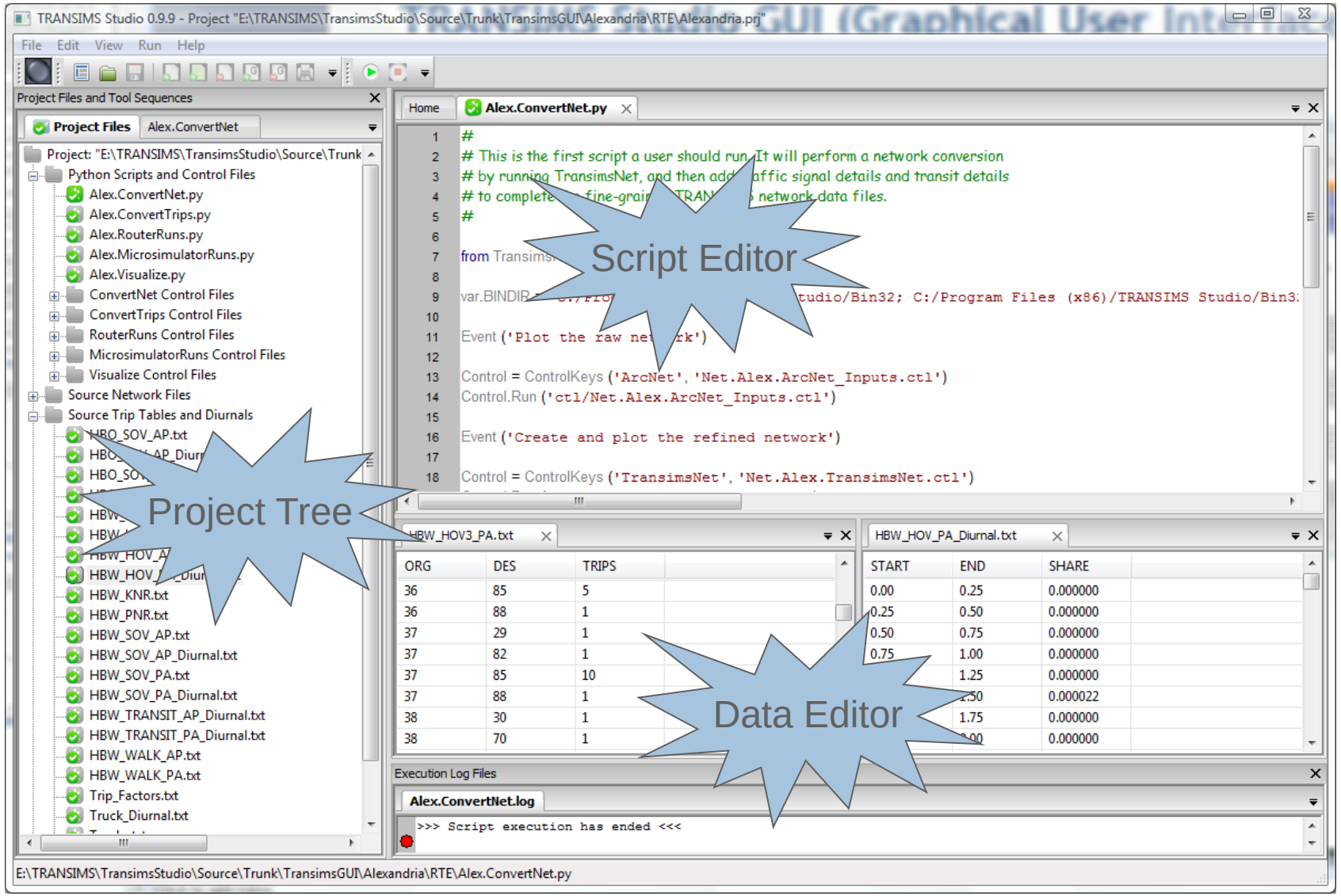
- TRANSIMS RTE is a Python Module that encapsulates the executables and makes scripting easy
  - Maximum flexibility for users by providing access to TRANSIMS from within one of the most powerful programming languages available today
  - Full caching of results and data files minimizes CPU time for restarts and alternative model runs
  - Syntax-aware control key objects create meaningful error messages and appropriate job termination upon errors
  - Multi-threading and remote execution allow parallelization through partitioning on an arbitrary number of CPUs and computers
  - Full cross-platform compatibility of scripts through virtualization of partitions and work queue execution
- While this all sounds fancy, it makes running models truly much easier – but the RTE targets advanced users and capabilities rather than novices
- Simplified mechanisms for beginners are being conceptualized

TRANSIMS Studio GUI (Graphical User Interface)

- TRANSIMS Studio has been developed based on modern integrated development environments such as Visual Studio
- The software allows access to all input and data files related to the model in a click and point interface
- Models are developed as Python scripts and are executed under the control of the GUI

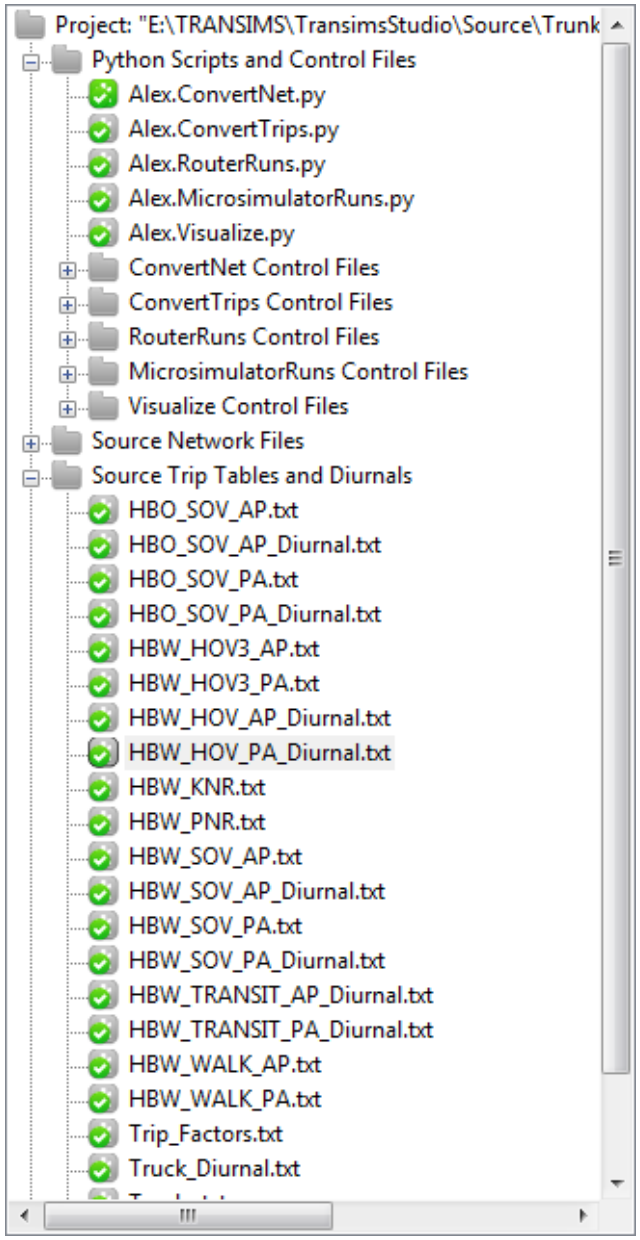


TRANSIMS Studio GUI (Graphical User Interface)



TRANSIMS Studio GUI - Project Files

- Existing files can be grouped by adding them from the disk
- Groups can be hierarchical and can be grouped in any arbitrary manner that is logical to the user
  - E.g., network files can be grouped together, or trip tables, or Python scripts
  - Order and grouping are completely up to the user
- The entries are basically simple references to files on the disk that can be easily accessed within the environment
- Basic SVN functionality provides version control if desired

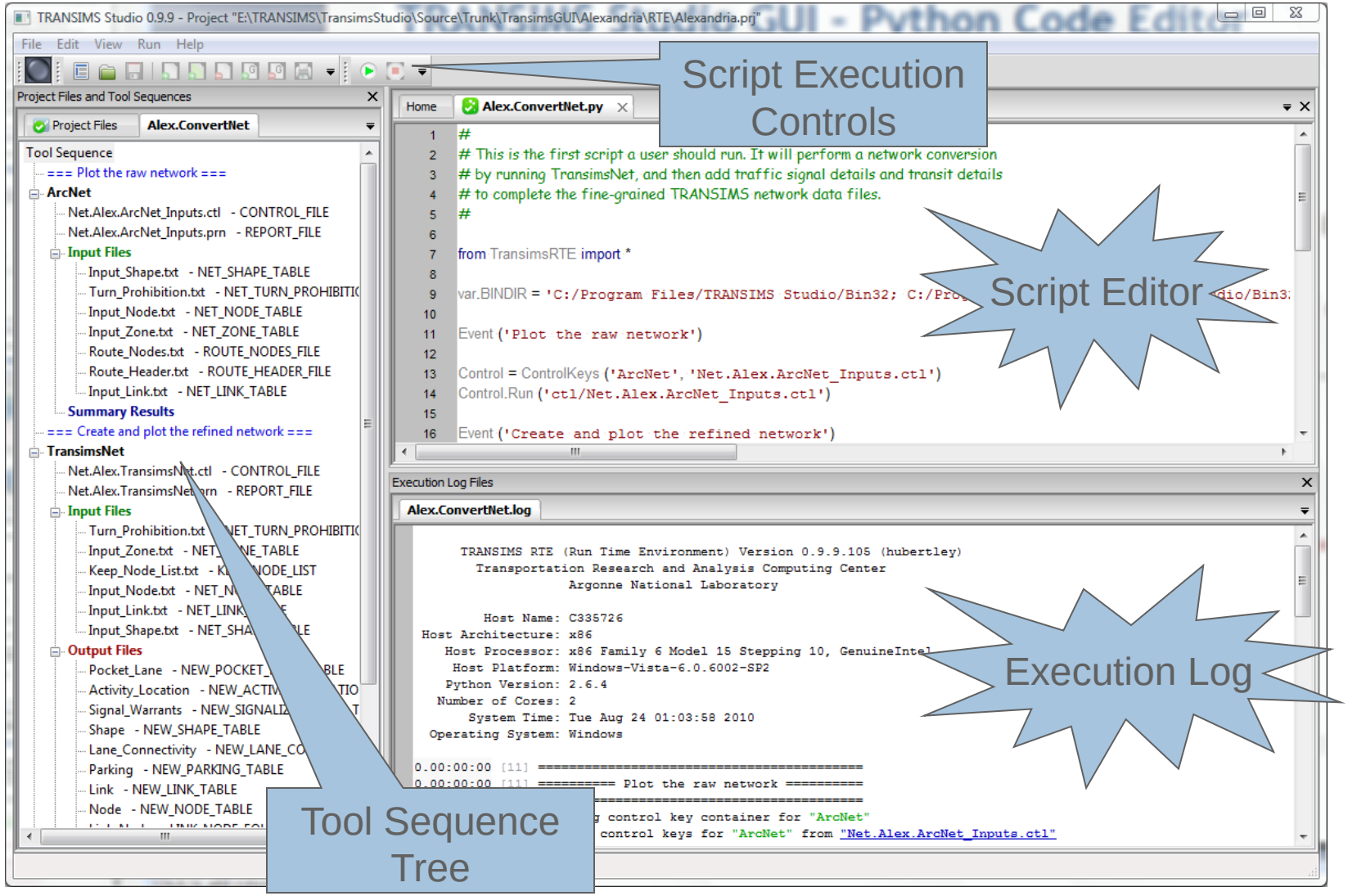


TRANSIMS Studio GUI - Python Code Editor

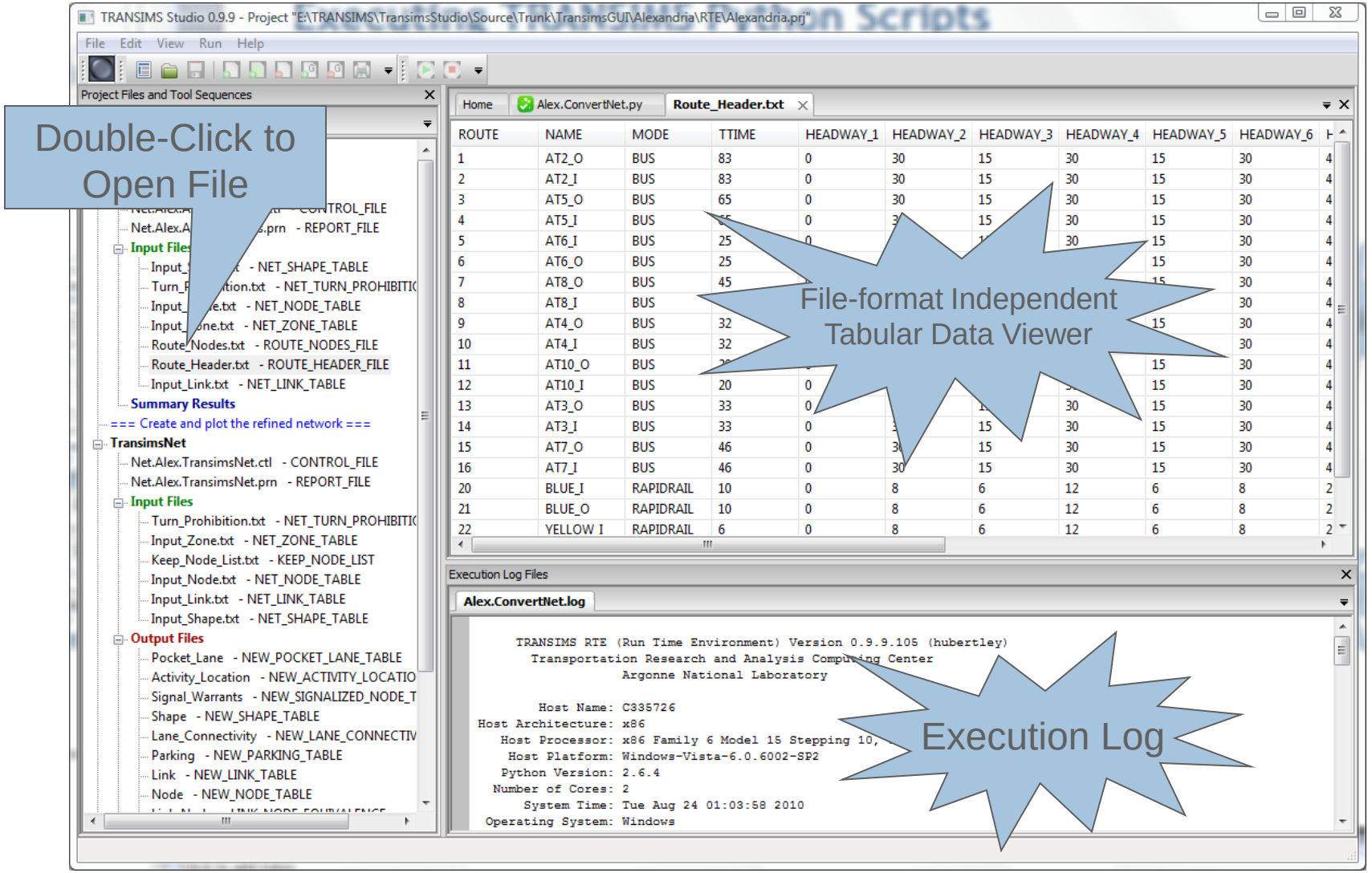
- A powerful code editor is included to simplify writing scripts
- Python syntax is color-coded to make programming easier
- Online help is accessible – e.g., select a key word and press F1 to get specific help
- The code window allows simple Python code to be written based on the RTE library

```
1  #
2  # This is the first script a user should run. It will perform a network conversion
3  # by running TransimsNet, and then add traffic signal details and transit details
4  # to complete the fine-grained TRANSIMS network data files.
5  #
6
7  from TransimsRTE import *
8
9  var.BINDIR = 'C:/Program Files/TRANSIMS Studio/Bin32; C:/Program Files (x86)/TRANSIMS Studio/Bin3
10
11 Event ('Plot the raw network')
12
13 Control = ControlKeys ('ArcNet', 'Net.Alex.ArcNet_Inputs.ctl')
14 Control.Run ('ctl/Net.Alex.ArcNet_Inputs.ctl')
15
16 Event ('Create and plot the refined network')
17
18 Control = ControlKeys ('TransimsNet', 'Net.Alex.TransimsNet.ctl')
```

Executing TRANSIMS Python Scripts



Executing TRANSIMS Python Scripts

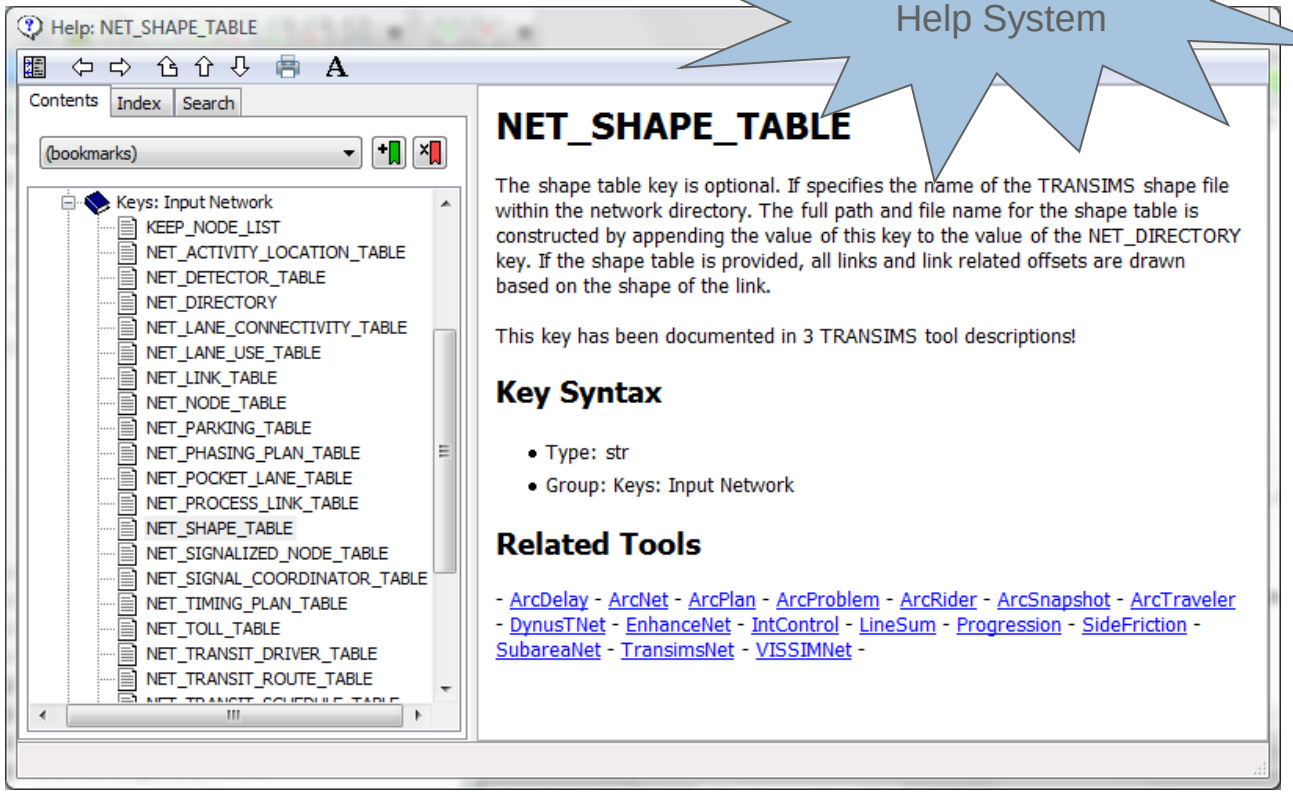


Context-Sensitive Help

```
5   #---- Input Files ----
6
7   NET_DIRECTORY                          ../inputs
8   NET_NODE_TABLE                         Input_Node.txt
9   NET_LINK_TABLE                         Input_Link.txt
10  NET_SHAPE_TABLE                        Input_Shape.txt
11  NET_ZONE_TABLE                         Input_Zone.txt
12
13  ROUTE_HEADER_FILE                      Route_Header.txt
14  ROUTE_NODES_FILE                       Route_Nodes.txt
15
```

Highlight and Press F1 to Open Help

Fully Cross-Referenced Help System



TransimsVIS Software Goals

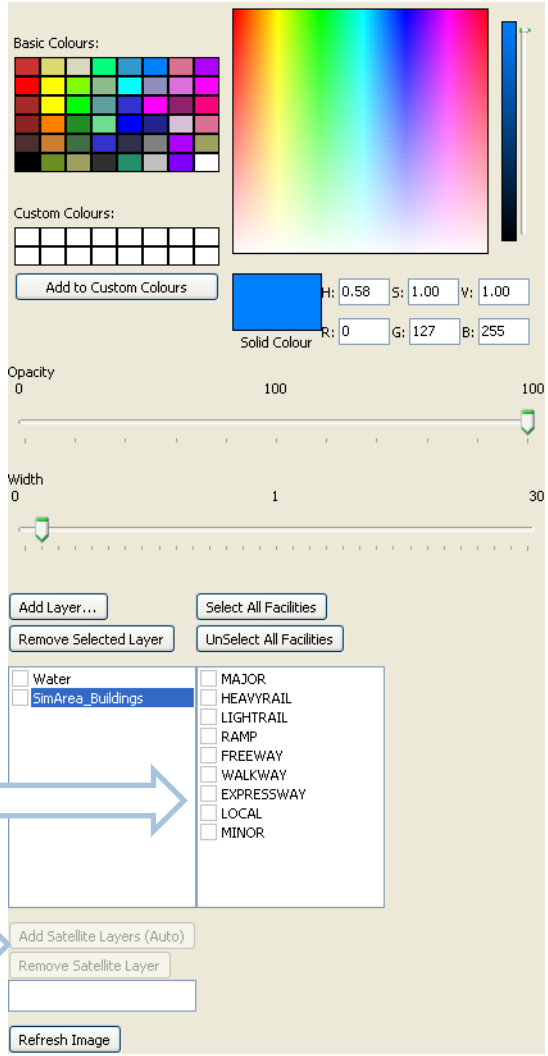
- TransimsVIS is designed to bridge the current shortcomings TRANSIMS has to easily display and decipher results once the program has run. The development was undertaken with several metrics in mind:

- **Backwards Compatibility**: The software would have the capability to display any of the current shape files being produced by TRANSIMS ArcUtilities
- **Interactivity**: Allow users to interact with snapshot (and other types of) data spatially and temporally
- **Movie Production**: Allow users to capture frames and render movie files for the purposes of presentation material
- **Creative Visualizations**: Provide creative paradigms for visualizing complex data concepts such as congestion
- **Cross Platform**: As TRANSIMS is cross platform, the visualization software should be as well. It will be integrated with the concurrently developed TRANSIMS Studio software package.

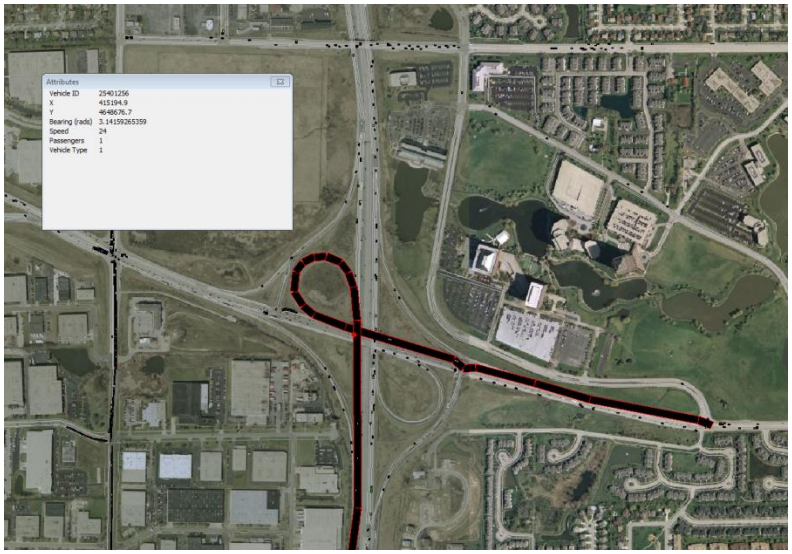
Backwards compatibility: ESRI ShapeFile Designer

- The software's interaction with shape files is handled through the open source program shp2img and the corresponding underlying .map files

- The Design Tab has several functions:
  - Changing Color and Opacity of Layers
  - Importing Shape Layers
  - Altering Color, Opacity, and Width of individual facility types
  - Importing Satellite Layers



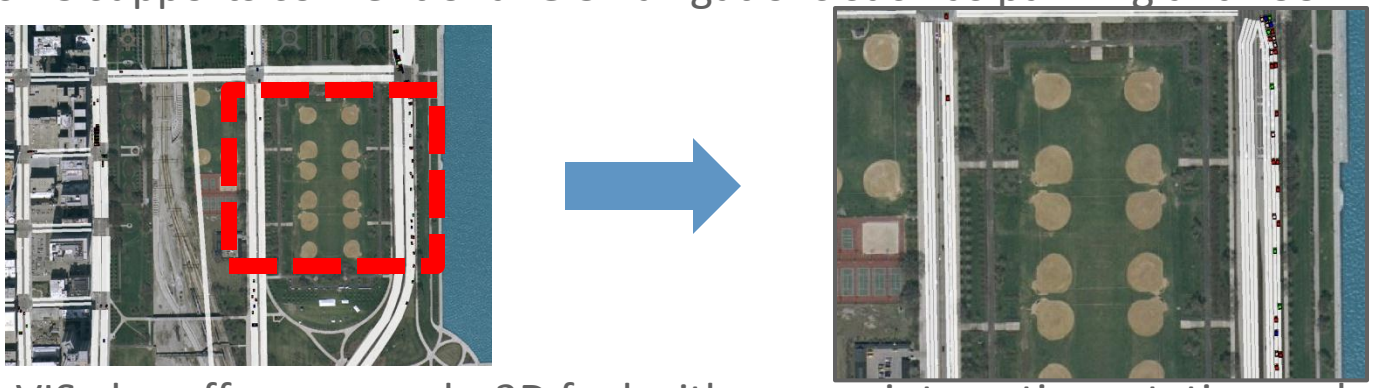
Backwards compatibility: GIS functionality



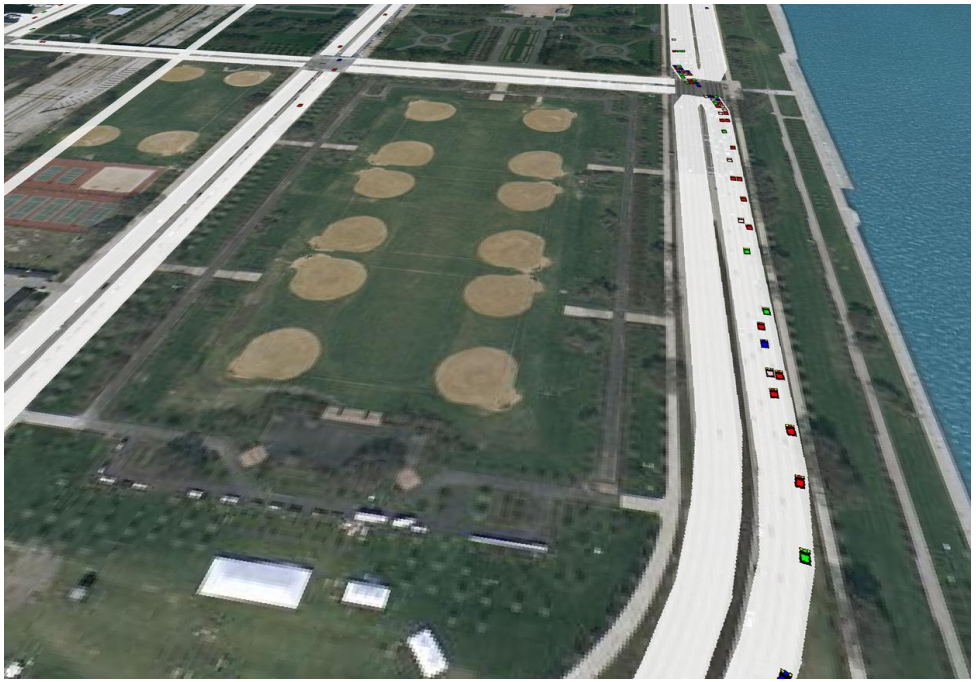
- An important aspect of visualization is to be able to provide a connection to the underlying attributes of a network through selection
- There are three enabled selection modes
  - Link
  - Node
  - Vehicle
- In addition, there is a compatibility with the underlying TRANSIMS plan file which allows a user to see a vehicle's full route upon selection

Interactivity: Navigation

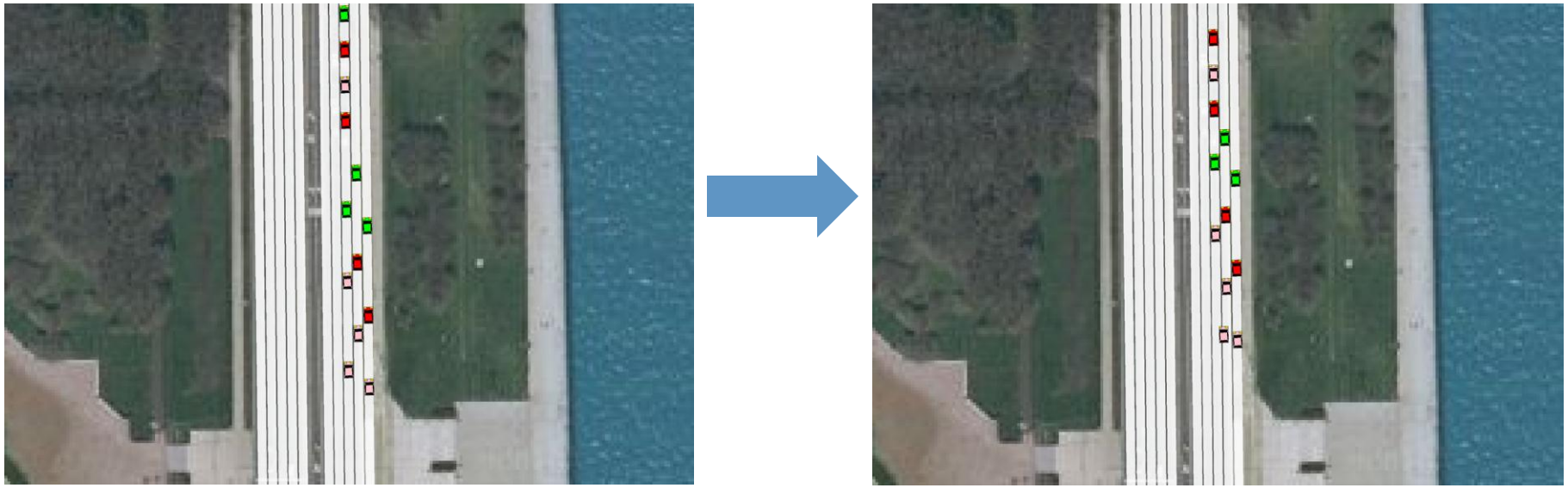
- TransimsVIS supports conventional GIS navigations such as panning and zooming



- TransimsVIS also offers a pseudo-3D feel with mouse interactive rotation and perspective transforms



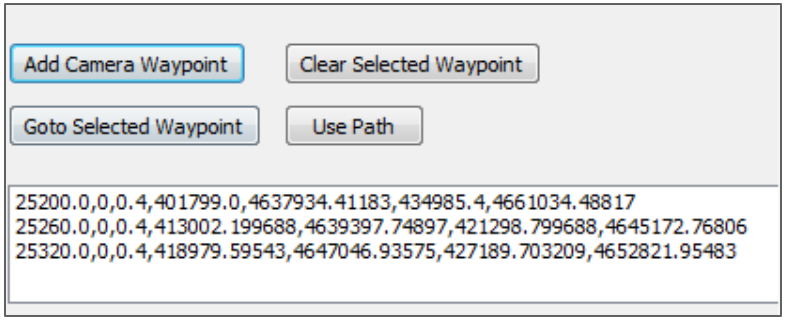
Interactivity: Visualizing Time-Dependant Data



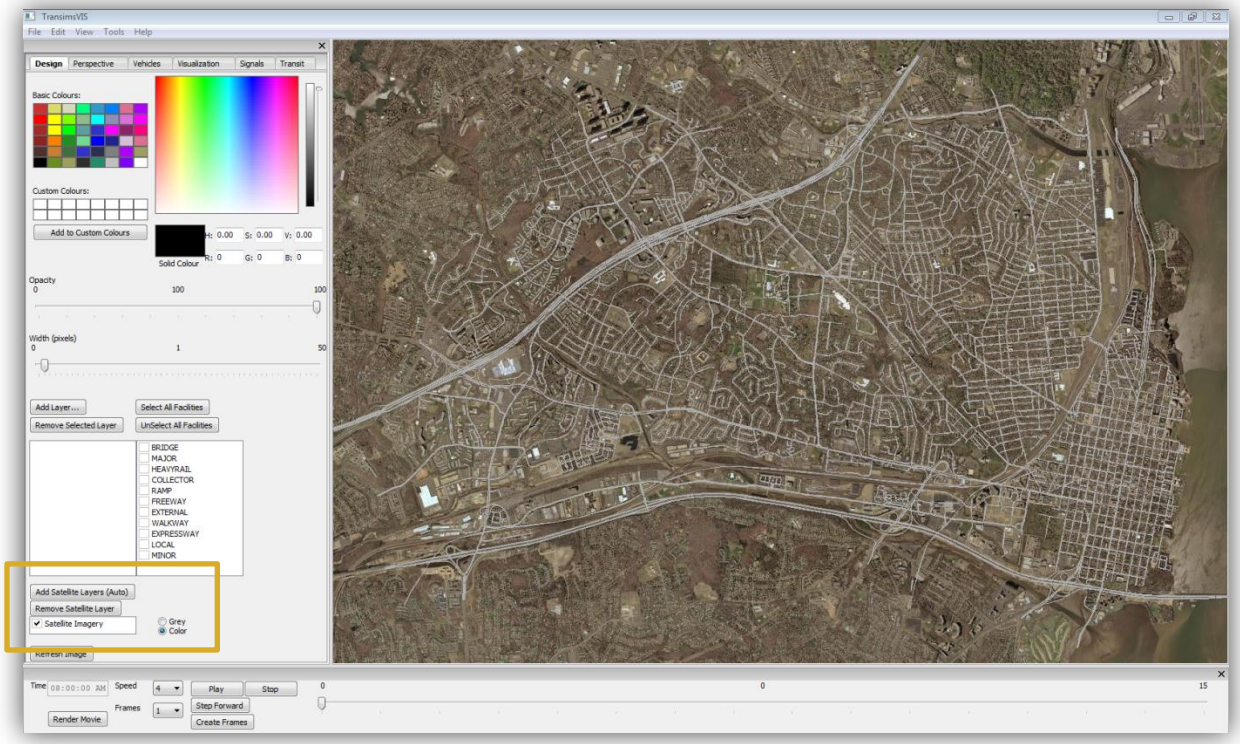
- There are several ways to navigate through time-dependant data with TransimsVIS
  - The time slider bar allows quick navigation between key frames
  - The user can step forward one frame at a time
  - The user can play the data at various speeds and varying levels of interpolation

Interactivity: Waypoint System

- A very popular feature in other mapping software is to save locations and move between them over time.
- The waypoint system available in TransimsVIS has the following features
  - Saving locations and times in a user-managed list
  - Returning to saved locations and times
  - Posting several locations and times which can be used during animation or movie production to create cinematic camera movements and multiple points of view



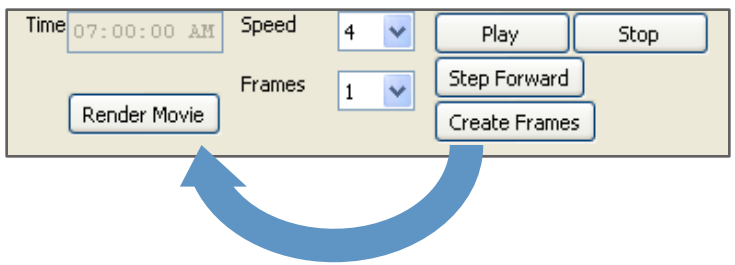
Interactivity: Satellite Imagery

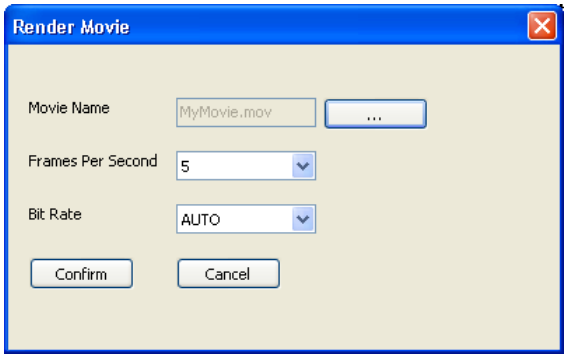


- TransimsVIS supports web-based acquisition of satellite imagery
- Routine connects to www.msrmaps.com (previously www.terraserver.com) and downloads satellite tiles
- These tiles are assembled on the fly and placed in back of the dynamic imagery canvas
- Tiles are cached in a repository on the hard drive so once they are downloaded, they need not be downloaded again

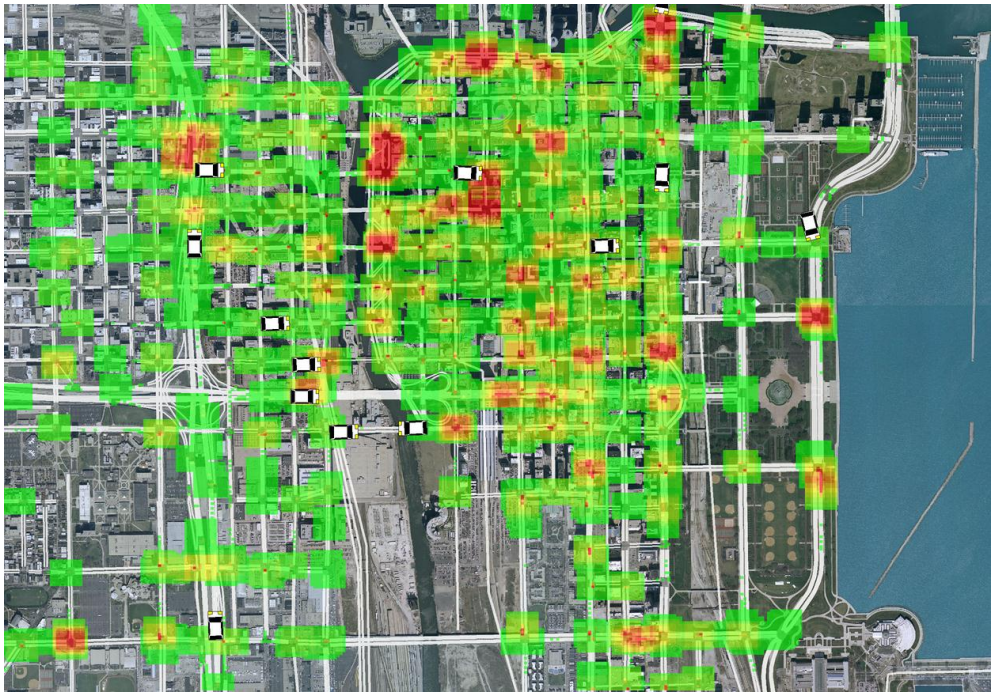
Movie Production

- Making a movie using TransimsVIS is fairly straightforward
  - Configure the interactive mode so that it looks as desired for the start point for your movie
  - Click the “Create Frames” button to begin generating frames – minimize the window or watch as each frame gets rendered
  - When the movie encompasses the time interval desired, click the “Stop” button
  - Use the “Render Movie” button to alter the movie name, FPS, and bit rate
  - Confirmation runs ffmpeg in the background to generate the movie requested



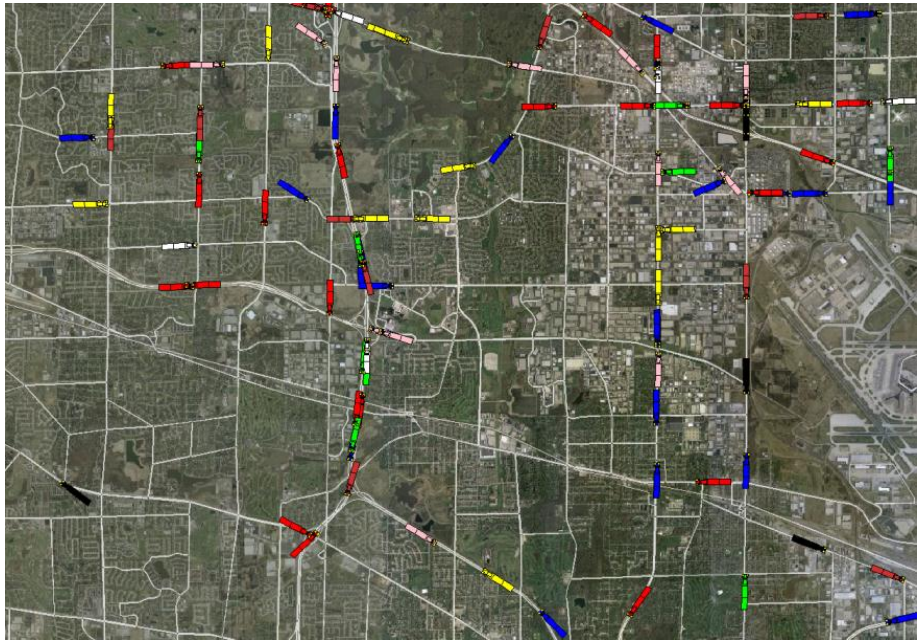


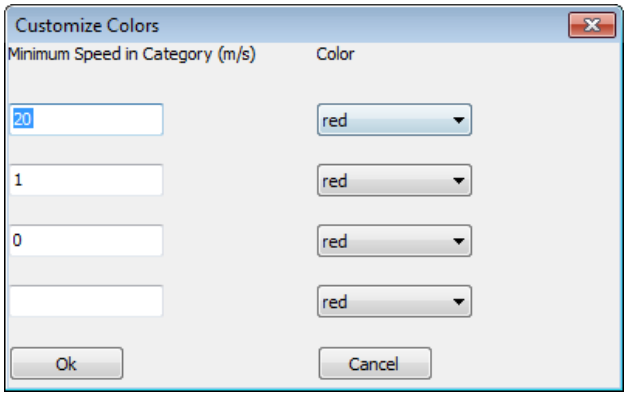
Creative Visualizations: Congestion Indicators



- The Heat Plot feature incorporates the following concepts:
  - Generates heat every step wherever a car is found.
  - Cars generate more heat the slower they moves.
  - Between time steps, the heat spreads to neighboring regions.
  - Over time a region will cool back down as it stops experiencing congestion.
- The Probe Vehicle feature incorporates the following concepts:
  - A random percentage of vehicles are selected to be “probe” vehicles
  - These cars are drawn larger so they are visible on the system level
  - This allows the user to easily understand what the average driver is feeling in various regions at one time

Creative Visualizations: Vehicle Categorization

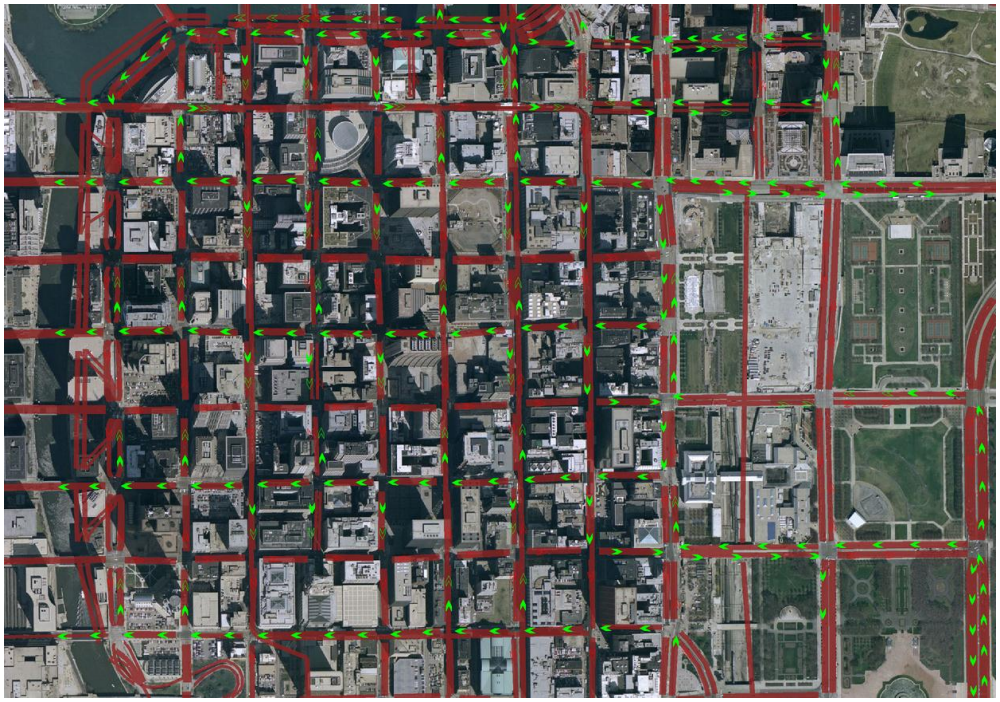




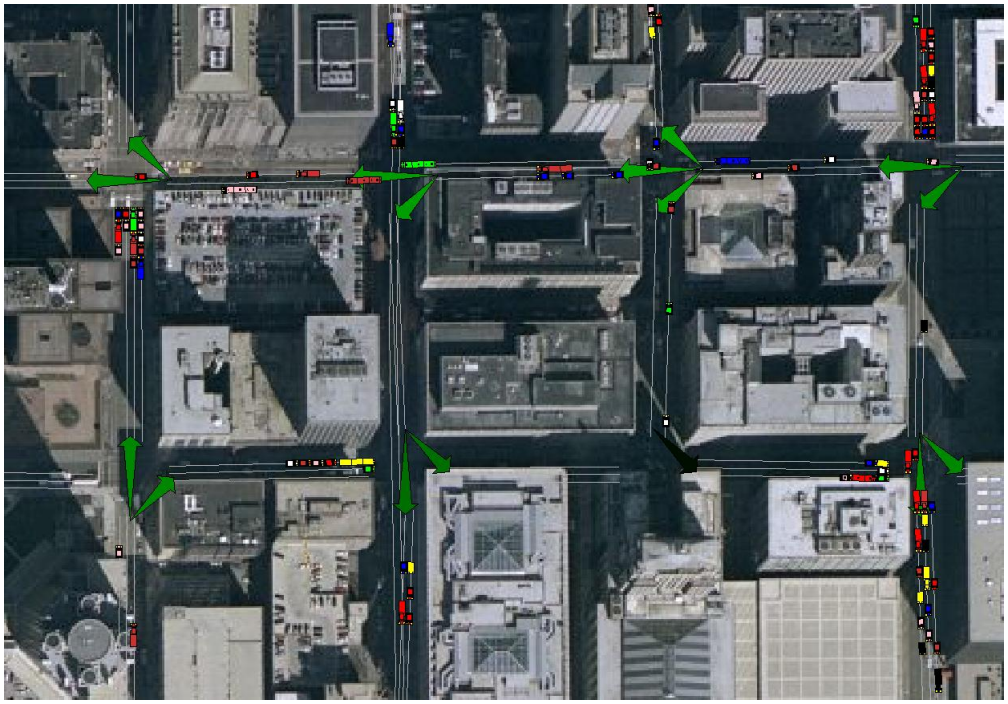
- TransimsVIS offers several options for categorizing vehicles and probe vehicles
  - Coloration by speed
  - Coloration by passengers
  - Coloration by vehicle type
  - Probe selection by percentage
  - Probe selection by vehicle type
  - Probe selection by input list

Creative Visualizations: Signal Synchronization Indicators

- Viewing signals at a system-level perspective is a very challenging task due to the ease of over-loading the visual system with all the possible information.
- The TransimsVIS system-wide signal viewer incorporates the following ideas:
  - Utilization of a conveyor technique to naturally direct the eyes to detect healthy synchronization patterns
  - Option of using arrows to indicate directionality for static plots
  - Fading and slowing of arrows as red signals are encountered



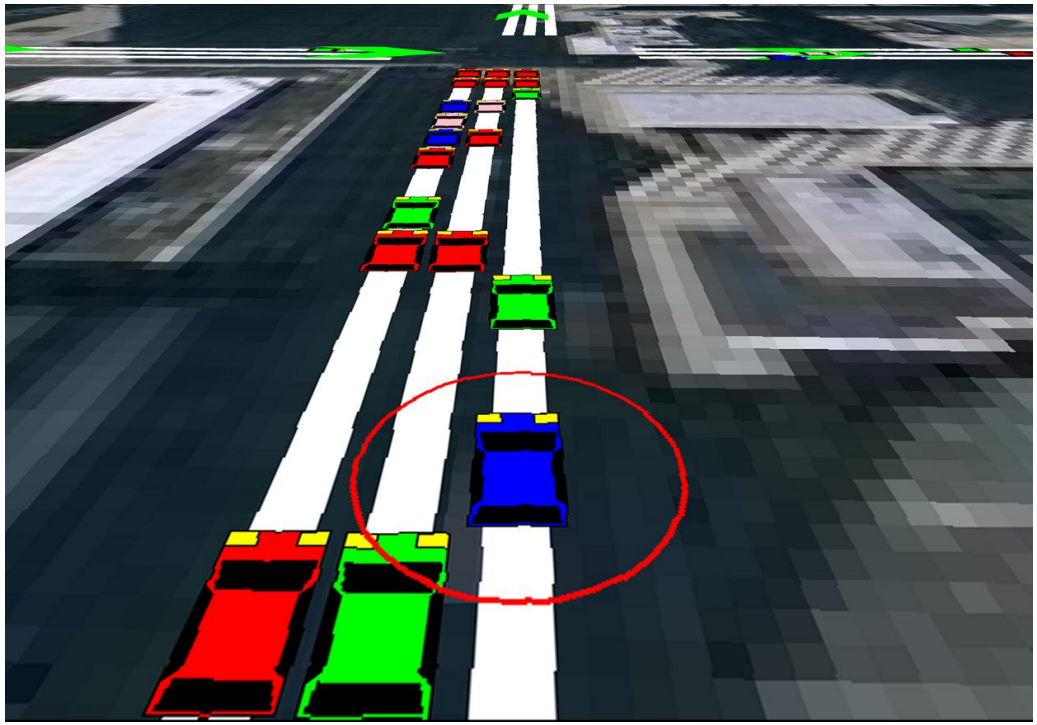
Creative Visualizations: Local Signal Systems



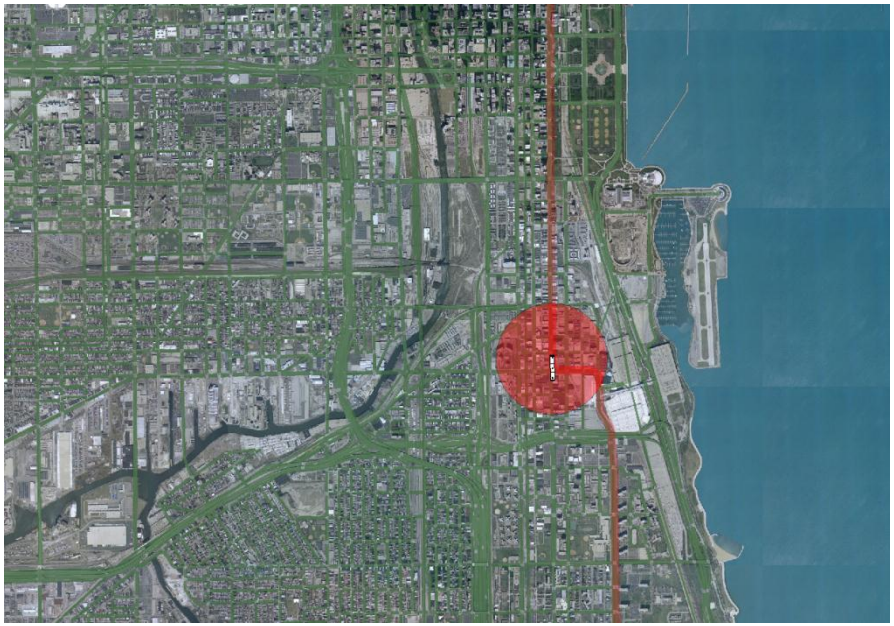
- Local signals can be enabled in addition to the regional signal paradigm
- These signals are used to evaluate operational characteristics in a visualization:
  - Arrows indicating which protected phases are enabled at a given time
  - Color shading based on amount of time remaining in a given phase (from green to black)
  - Full integration with the TRANSIMS system event file for accurate depiction of actuated signals

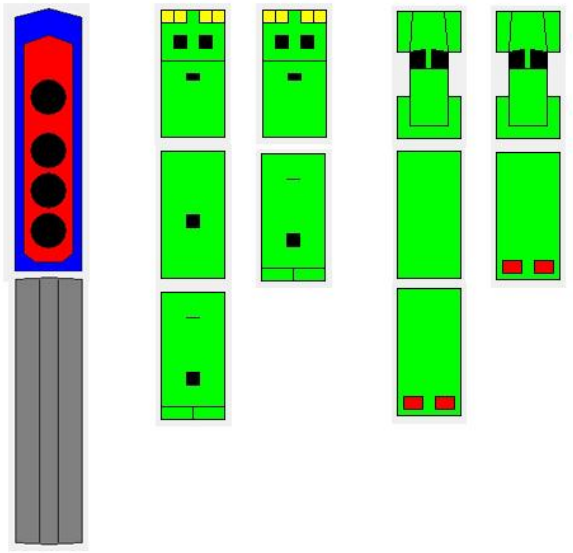
Creative Visualizations: Driver's Perspective

- TransimsVIS offers the following technique to allow the user to see what the average driver experiences on their trip
  - Zoom in and rotate to nearly the driver's perspective
  - Attach the camera to a car
  - Follow the car second by second (or finer) through a portion of their trip



Creative Visualizations: Transit and Trucks

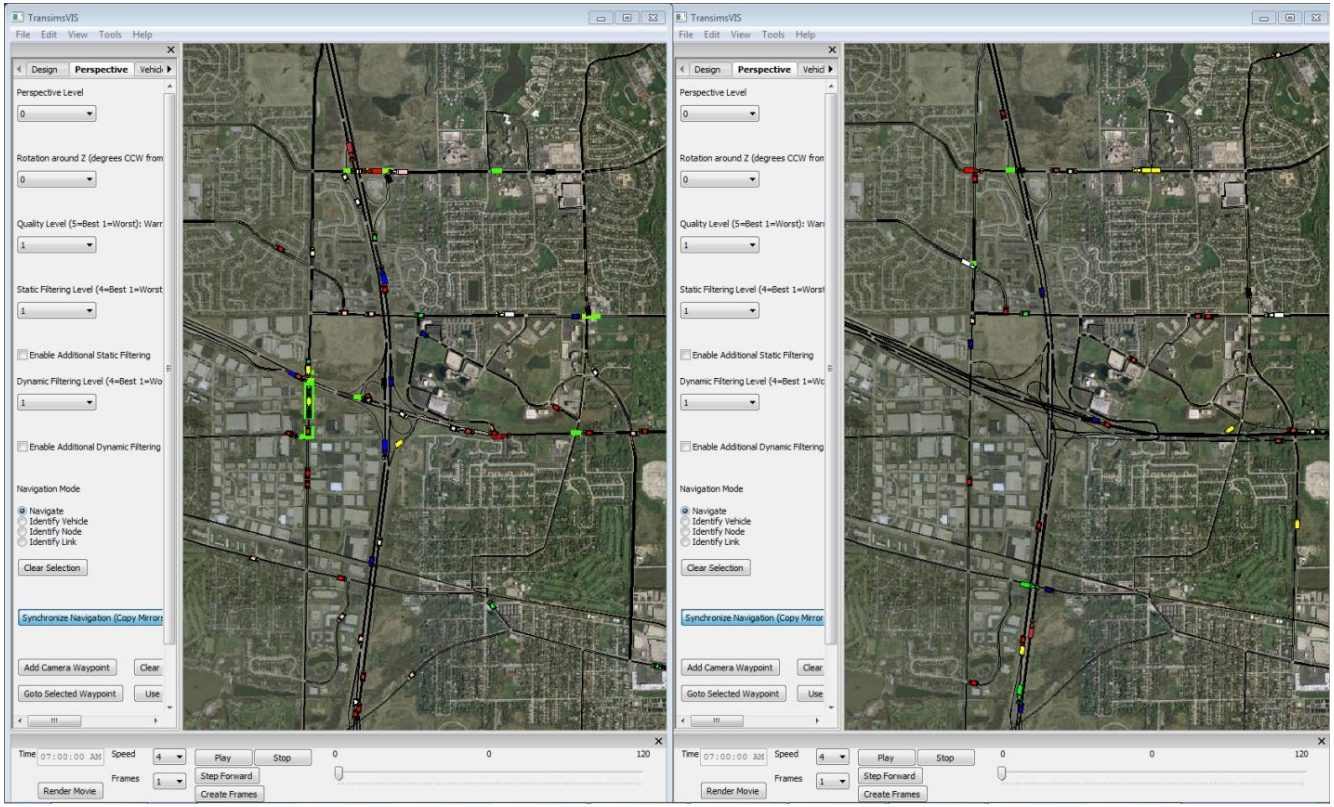




- Currently TransimsVIS supports three Transit-specific visualizations and has special modular 2D models for trains, buses, trucks
  - Display of full route upon selection of the vehicle
  - A circle displaying the relative ridership to other transit vehicles being drawn
  - Coloration of circle based on passenger volume categories

Creative Visualizations: Case Comparisons

- TransimsVIS offers a capability to spawn a copy of the visualizer to be used in case comparisons and synchronize the navigation between the two copies. Paired with the ability to save configuration files, this becomes an effective way to perform a case study.



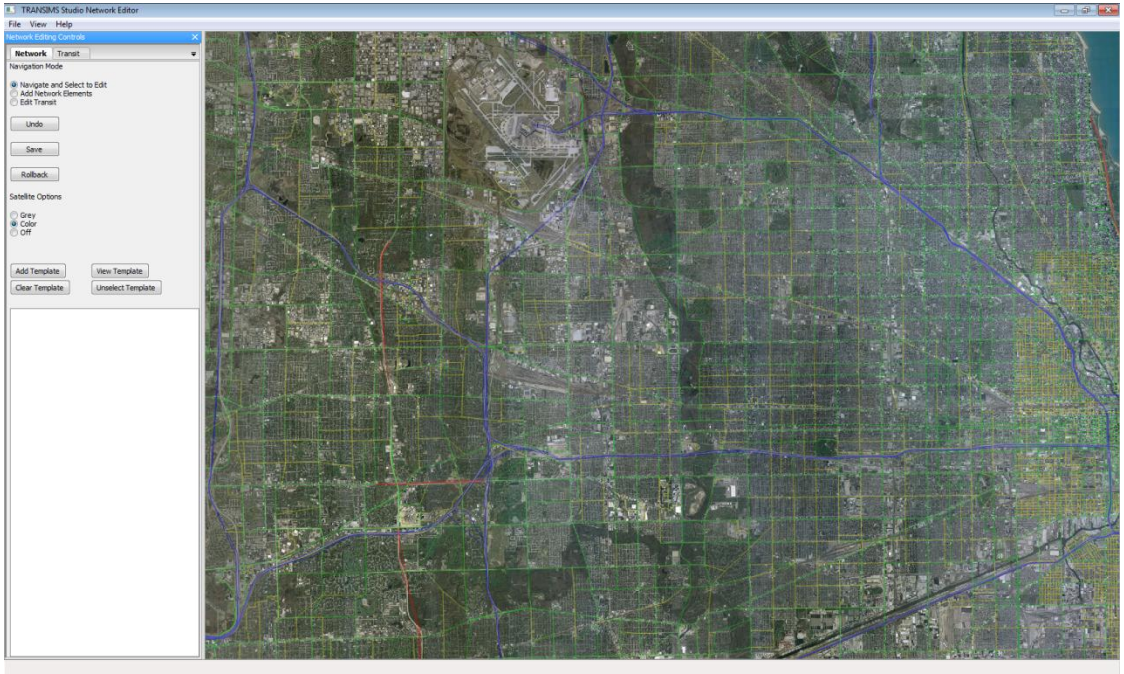
TransimsEDT - *Software Goals*

- Network Editing is a vital element of transportation modeling facilitating the construction of a traffic network which matches the real world as closely as possible

- TransimsEDT development focused on the following improvements:
  - Installation – available as a “drop-on-desktop” executable or traditional installer
  - Performance – greatly improved satellite system and editing operations
  - User Interface – surveyed users to develop the most intuitive editing keyboard/mouse shortcuts
  - Consistency with TRANSIMS Studio Open Source paradigm
  - Written in Wx Python for fast and flexible development

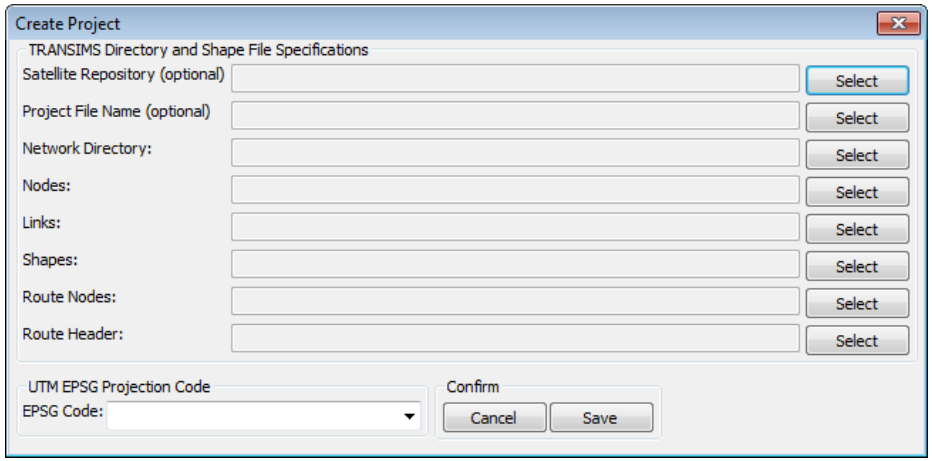
TransimsEDT - *Performance*

- Efficient network drawing algorithm
  - 4-tiered drawing algorithm allowing optimal network draw speed across all editing modes, zoom levels, and platforms
- Loading of satellite imagery greatly improved
  - Web-based acquisition of satellite tiles
  - Hard drive caching of previously fetched satellite tiles for quick access in the future
  - Efficient image assembly algorithm



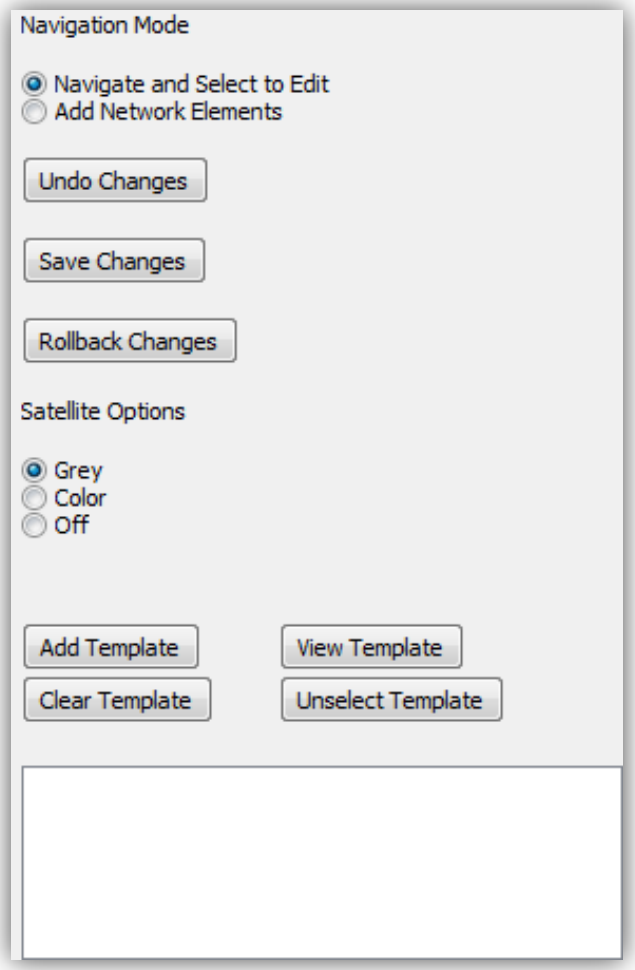
TransimsEDT - *General Features*

- Project file constructible by menu
  - Utilizes TRANSIMS ".ctl" format for modification later
  - Allows quick project save/load
- Full "undo" system
  - Intelligent undo system determines when to undo a group of actions
  - All edits are able to be undone
- Save states
  - Save intermediate state in memory and rollback to it later
  - When finished can export a hard copy of network tables to TRANSIMS format

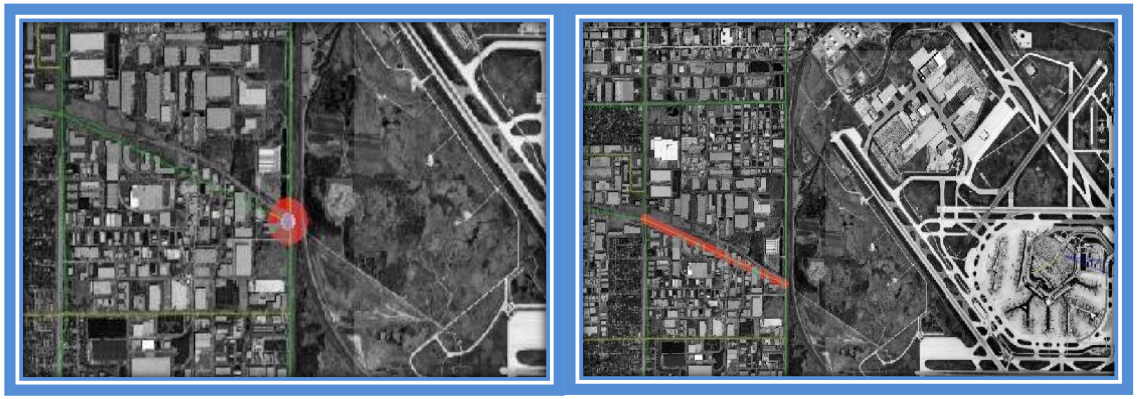


TransimsEDT - *Network Editing Controls*

- **Navigation Mode:**
  - **Modes** – redefines hotkeys and mouse operations based on mode selected
- **Data Options:**
  - **Undo Changes** - removes last change
  - **Save Changes** – saves changes in memory
  - **Rollback Changes** – removes all changes made since last save
- **Satellite Options:**
  - **Options** – select for coverage and performance
- **Template System:**
  - **Add Template** – creates a template of node/link
  - **Clear Template** – removes template
  - **Unselect Template** – previously selected template no longer highlighted
  - **View Template** – examine template



TransimsEDT - *Navigate and Select to Edit*

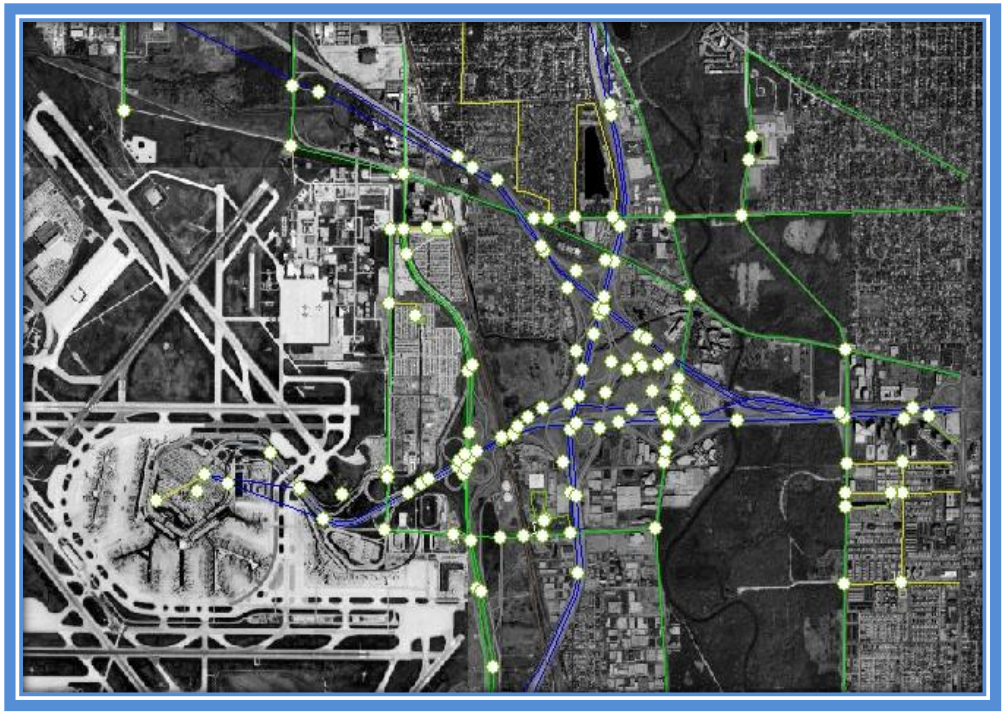


Select To Edit

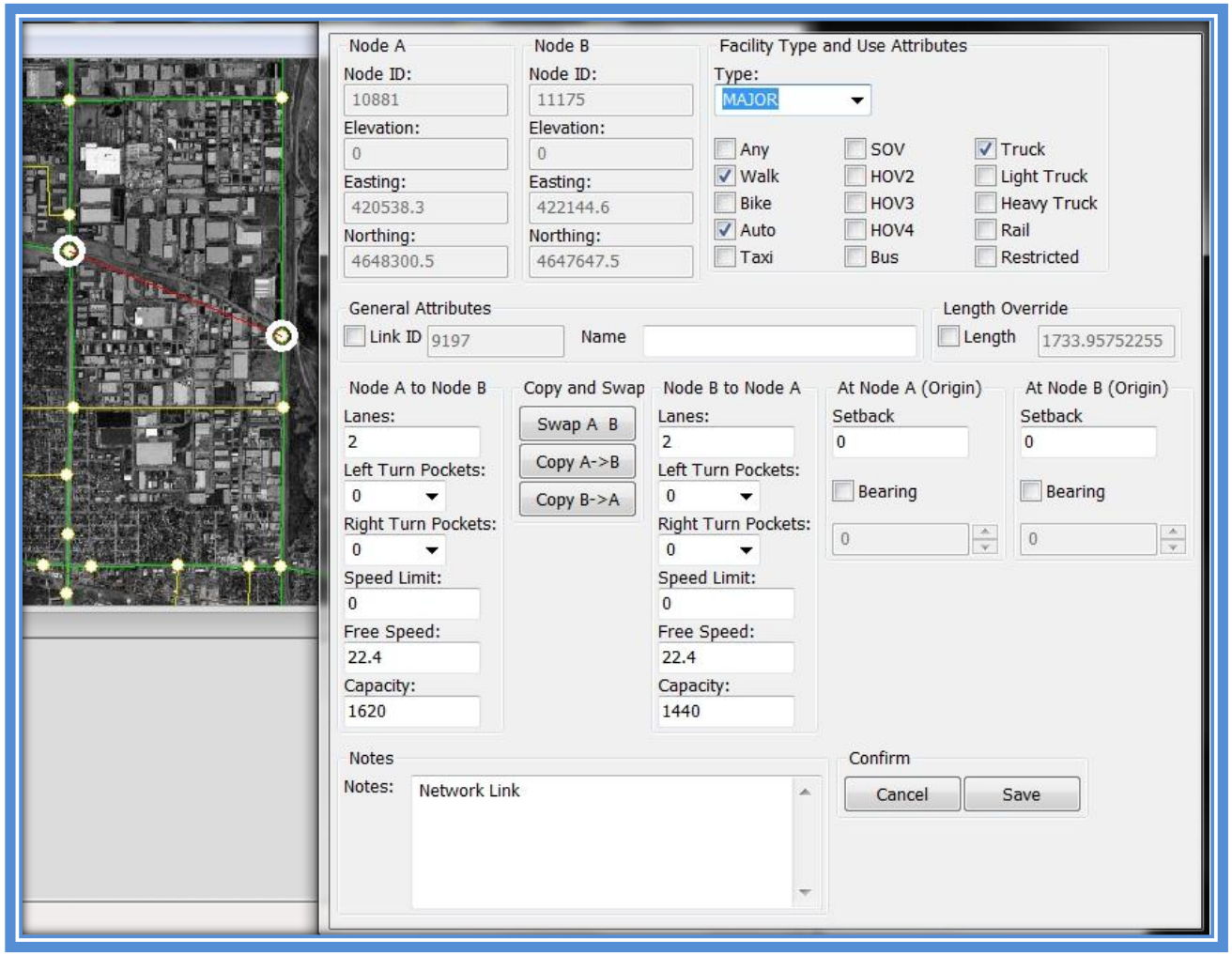
- Clicking a link or node initializes edit mode

Navigate

- Use mouse
  - Left click and hold to pan
  - Scroll wheel to zoom
- Use arrow keys to pan
  - Left ←
  - Right →
  - Up ↑
  - Down ↓
- Use **(= or +) / -** to zoom
- Mouse-over to highlight node/link



TransimsEDT - *Navigate and Select to Edit*



Select to edit

- View Link Attributes
  - double click on link
- View Node Attributes
  - double click on node
- Reshape Link
  - left click on link and backspace key
- Move shape point or intersection
  - left click, hold, move and release

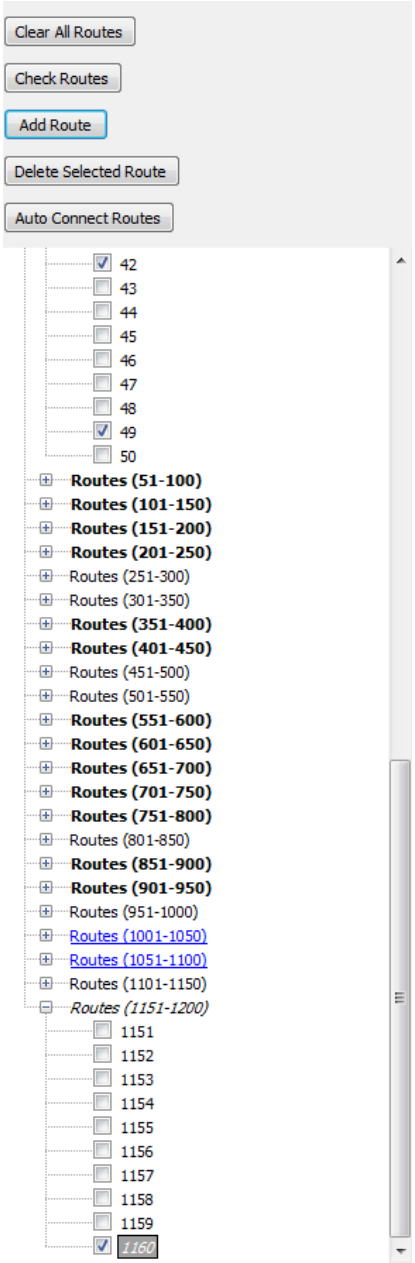
Add Network Elements

- Link Drawing
  - Left Click on Map or Node to start drawing a Link
  - Extend link with mouse, add shape points with left click
  - Terminate link with “Enter” or by clicking on an existing node
- Shape Modification
  - Left Click on existing link to add a shape point at that point
  - Use mouse to move shape point and left click to place
- Link Splitting
  - Ctrl + Left Click on existing link to split it
  - When drawing link, Ctrl + Left Click on existing link to split it and terminate link at that point

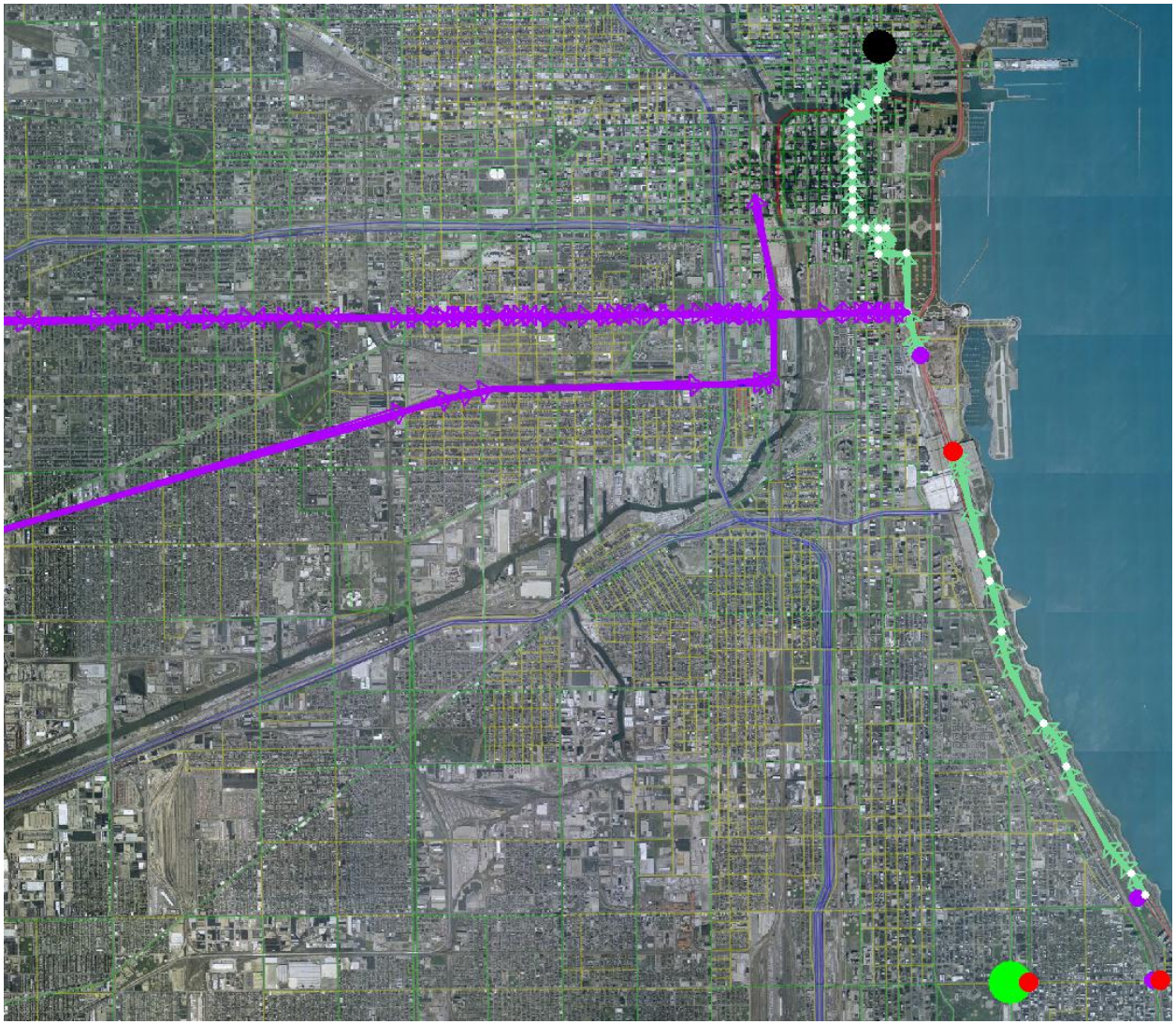
TransimsEDT - Transit Editing

Transit Editing Menu

- Check which routes are damaged, invalid, or incomplete
- Add or Delete a route
- Auto-Correct Routes using mini-routing routine
- Route Display Box
  - Double click to zoom to route
  - Check to display
  - Bold – Damaged
  - HyperText – Autocorrected
  - Italicized - New



TransimsEDT - Transit Editing



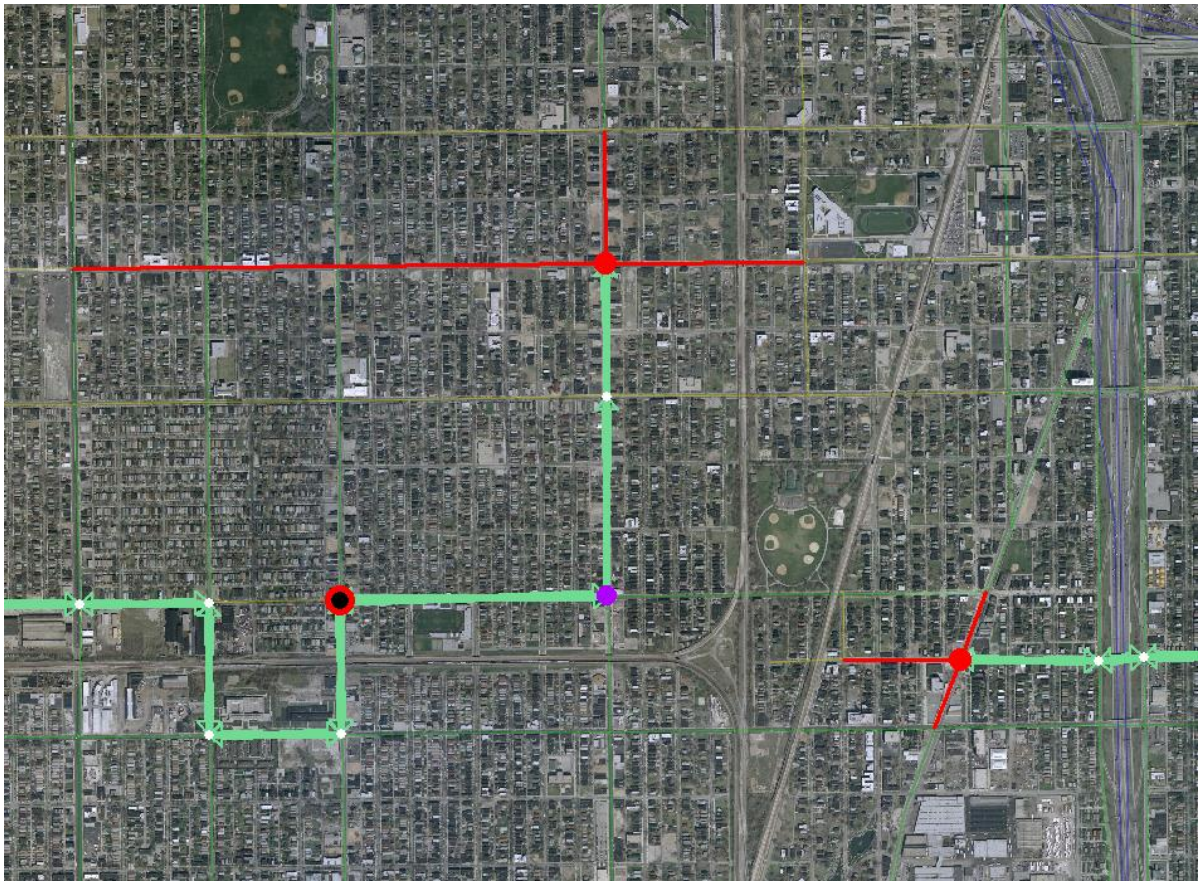
Transit Route Display

- Selected Route in Aqua, others in Purple
- Green circle indicates the route starting point, Black circle indicates route ending point
- Red circle indicates start of route break, Purple circle indicates end of route break

TransimsEDT - Transit Editing

Transit Route Building

- Red links indicate suggestions for Transit Accessible path choices
- Left click on link to extend route path to that link
- Left click on node (Red circle with black circle) to select node for deletion



Credits and Acknowledgements

- GIS visualization materials were mostly developed at Argonne based on the TRANSIMS tools developed by AECOM for USDOT
- Chicago road and transit network data used in some of the examples was provided by the Chicago Metropolitan Agency for Planning
- USDOT provided the funding for the development of these training materials
- USDOT provided the funding for the TRACC computing center and the resources necessary to perform these training session
- Some figures have been developed for USDOT by Prof. Antoine Hobeika, Virginia Polytechnic Institute, Civil and Environmental Engineering
- The presentation is in part based on materials provided by USDOT at a training course in November 2006