Argonne NATIONAL LABORATORY

TRANSIMS Version 5 Introduction

January 20, 2011

David Roden – AECOM

AECOM

Topics

- User feedback
- Desired improvements
- Guiding principals
- Global changes
- Network files
- Demand files
- Version 4 compatibility

User Feedback

- User interface concerns
  - Network files are too cumbersome for efficient editing
    - Version 3/4 field names, multi-file/record dependencies,…
  - It is too easy to introduce errors in control files
    - Inconsistent key names, units of measure, key groups,…
  - Plan file processing and sorting problems
    - Node/link, traveler scaling, multi-leg trips, time/traveler sort,…
  - Partitioning difficulties
    - File extensions vs. command lines, aggregate statistic reports,…
  - How to link tools into modeling algorithms
    - Router/Microsimulator stabilization, user-equilibrium convergence,…
  - GUI tools for editing, running and visualizing

Desired Improvements

- Functionality and performance needs
  - A higher fidelity and scalable Microsimulator is needed
    - Cell-based speeds, lost vehicles, signal coordination, …
    - Single processor limitations – simulation size and processing time
  - Better coordination between Router and Microsimulator
    - Plan leg scheduling issues, transit options, on-the-fly re-routing,…
  - Path attributes to support other models/software
    - Forward and backward path building (time control points)
    - One-to-many skims without creating plan files
    - Linkable routing service class/subroutine
  - The custom data classes are too complicated for new programmers to quickly build upon
    - Needs to be easier to learn/use with fewer/no variations/overrides

Guiding Principles – User Help

- Simplify editing
  - Simplify the network coding requirements
  - Reduce the number of coded dependencies between files
  - Use data nesting to avoid sorting problems and record inconsistencies
- Reduce user errors
  - Provide more program-based help information
  - Standardize control keys and key definitions
  - Interpret user-provided unit specifications

Guiding Principles – Performance

- Enhance performance
  - Multi-threading and multi-processor options
  - Streamline the Router → Plan Processing → Microsimulator interaction
  - Enhance the Router and Microsimulator functionality and fidelity
- More programmer friendly
  - Standard Template Library
    - strings, streams, vectors, maps, etc.
  - Centralize codes, standardize and automate processing
  - Create DLL services for linkages to other software

Global Changes

- Units of measure
- Time data and formats
- Control key processing
- Configuration files
- Command line helps
- Definition files
- Partitions
- Subareas

Units of Measure

- Global control key
  - UNITS_OF_MEASURE = ENGLISH or METRIC
    - Defaults to metric for backward compatibility
  - Used for all output files and printouts
- Input units
  - Units specified in control keys and *.def files
    - Defaults to Version 4.0 usage
  - Automatically converts units from one system to the other
- Consistent internal units
  - Feet (meters), feet/second (meters/second), seconds and cents
  - One decimal point (e.g., 52.1 feet, 25.2 fps, 10.1 seconds, 5.2 cents)

Time Data and Formats

- New data object: Dtime – day-time
  - Integer tenths of seconds (time steps)
  - Stored in files using the “TIME” data type
    - In Binary files = 2 or 4 byte integer
    - In Text files = ~12 character string
- Time formats
  - Seconds, Minutes, Hours, Hour Clock, Day Time, and Time Code
  - Individual control keys can include time units (e.g., 15 minutes)
- Global control keys
  - TIME_OF_DAY_FORMAT = DAY_TIME (1@12:34:56.7)
  - MODEL_START_TIME = 0:00
  - MODEL_END_TIME = 24:00

Multiple days and start times other than midnight (e.g., 3:00 AM)

Control Key Processing

- Object-oriented key processing
  - Standard output, unit conversion, range checking, error messages
  - Control keys and key values (except file names) are not case sensitive
- Key data structure
  - Key number, name, nesting level, optional/required status, data type, default value, data range, and help index
    - DEFAULT_FILE_FORMAT, "DEFAULT_FILE_FORMAT", LEVEL0, OPTIONAL, TEXT_KEY, "TAB_DELIMITED", FORMAT_RANGE, FORMAT_HELP
    - ACTIVITY_DURATION, "ACTIVITY_DURATION", LEVEL1, OPTIONAL, TIME_KEY, "0.0 hours", "0.0, 0.25..24.0 hours", NO_HELP
- Key services support multiple programs
  - Execution, File, Data, Select, Flow-Time, Projection, Router, Simulator

Configuration File

- A configuration file can set global control keys
  - Set TRANSIMS_CONFIG_FILE=…

```
PROJECT_DIRECTORY                 ../
DEFAULT_FILE_FORMAT               TAB_DELIMITED
TIME_OF_DAY_FORMAT                HOUR_CLOCK
MODEL_START_TIME                  0:00
MODEL_END_TIME                    27:00
UNITS_OF_MEASURE                  ENGLISH
NUMBER_OF_THREADS                 4
NOTES_AND_NAME_FIELDS             TRUE
```

- Program control keys override configuration keys

Command Line Helps

- Command line flags
  - Program [-flag -flag] [control filename] [partition number]

-Q[uiet]	execute without screen messages
-H[elp]	show program syntax and control keys
-C[ontrol]	create/update a default control file
-K[eyCheck]	list unrecognized control file keys
-P[ause]	pause before exiting
-N[oPause]	never pause before exiting
-D[etail]	execute with detailed status messages
-X[ML]	write an XML file with control keys

- Example:

```
Router –q –k –p Router.ctl 0
```

Help Flag (-h)

- Displays control key name, status, data type, and default value for all keys recognized by the program

```
Control File Keys:
  DEFAULT_FILE_FORMAT             Opt.Text = TAB_DELIMITED
  TIME_OF_DAY_FORMAT              Opt.Text = DAY_TIME
  MODEL_START_TIME                Opt.Time = 0:00
  MODEL_END_TIME                  Opt.Time = 24:00
  UNITS_OF_MEASURE                Opt.Text = METRIC
  DRAW_NETWORK_LANES              Opt.Bool = FALSE
  LANE_WIDTH                      Opt.Dec. = 3.5 meters
  CENTER_ONEWAY_LINKS             Opt.Bool = FALSE
  LINK_DIRECTION_OFFSET           Opt.Dec. = 0.0 meters
  DRAW_AB_DIRECTION               Opt.Bool = FALSE
  POCKET_SIDE_OFFSET              Opt.Dec. = 2.0 meters
  PARKING_SIDE_OFFSET             Opt.Dec. = 3.0 meters
```

Control Flag (-c) (-cx)

- Create or updates a control file
  - If file name does not exist, create and populate with default values
  - If file name does exist, update the file by adding help messages to existing keys and adding default values for other keys
    - -C updates the file and continues processing
    - -CX updates the file and exits the program without processing

```
DEFAULT_FILE_FORMAT        TAB_DELIMITED      //---- TEXT, BINARY, FIXED_COLUMN, …
TIME_OF_DAY_FORMAT         HOUR_CLOCK         //---- SECONDS, MINUTES, HOURS, …
MODEL_START_TIME           0:00               //---- >= 0 [seconds], 0.0 [hours], 0:00
MODEL_END_TIME             27:00              //---- > [model_start_time]
UNITS_OF_MEASURE           ENGLISH            //---- METRIC, ENGLISH
DRAW_NETWORK_LANES         FALSE              //---- TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
LANE_WIDTH                 3.5 meters         //---- 0..40 meters
CENTER_ONEWAY_LINKS        FALSE              //---- TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
LINK_DIRECTION_OFFSET      0.0 meters         //---- 0..50 meters
DRAW_AB_DIRECTION          FALSE              //---- TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
POCKET_SIDE_OFFSET         2.0 meters         //---- 0..82 meters
```

XML Flag (-x)

- Creates a file:
  - ControlName.ctl → ControlName.xml
  - if no control file is provided → Program.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<TRANSIMS>
<PROGRAM NAME="ArcNet" VERSION="5.0.0" COPYRIGHT="2010 by TRANSIMS Open-Source" />
<CONTROL_KEYS>
<KEY CODE="404" NAME="DEFAULT_FILE_FORMAT" REQUIRED="false" TYPE="Text"
DEFAULT="TAB_DELIMITED" RANGE="TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,
SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, SQLITE3, VERSION3"
VALUE="TAB_DELIMITED" />
<KEY CODE="405" NAME="TIME_OF_DAY_FORMAT" REQUIRED="false" TYPE="Text"
DEFAULT="DAY_TIME" RANGE="SECONDS, MINUTES, HOURS, HOUR_CLOCK, DAY_TIME, TIME_CODE"
VALUE="HOUR_CLOCK" />
<KEY CODE="5" NAME="LANE_WIDTH" REQUIRED="false" TYPE="Decimal" DEFAULT="3.5 meters"
RANGE="0..40 meters" VALUE="3.5 meters" />
<KEY CODE="6" NAME="CENTER_ONEWAY_LINKS" REQUIRED="false" TYPE="Bool" DEFAULT="FALSE"
RANGE="TRUE/FALSE, YES/NO, 1/0, T/F, Y/N" VALUE="FALSE" />
</CONTROL_KEYS>
</TRANSIMS>
```

Definition Files (*.def)

- Specifies software version and field units
  - Binary files use field codes rather than strings

```
TRANSIMS50, TAB_DELIMITED, 2, NESTED
PARKING, INTEGER, 1, 10
LINK, INTEGER, 2, 10
DIR, INTEGER, 3, 1
OFFSET, DOUBLE, 4, 8.1, FEET
TYPE, STRING, 5, 10, PARKING_TYPE
NUM_NEST, INTEGER, 6, 2, NEST_COUNT
USE, STRING, 1, 128, USE_TYPE, NESTED
START, TIME, 2, 16, HOUR_CLOCK, NESTED
END, TIME, 3, 16, HOUR_CLOCK, NESTED
SPACE, UNSIGNED, 4, 5, NO, NESTED
TIME_IN, TIME, 5, 12, SECONDS, NESTED
TIME_OUT, TIME, 6, 12, SECONDS, NESTED
HOURLY, UNSIGNED, 7, 5, CENTS, NESTED
```

Partitions

- All demand files can be partitioned
  - household.*, vehicle.*, trip.*, selection.*, plan.*, problem.*, skim.*
- Version 5 uses number extensions rather than letters
  - *.AA , *AB, … and *.tAA, *tAB, … → *.0, *.1, …
- Programs process all partitions or a single partition
  - Partition number included on the command line
- Each partition uses an independent execution thread
  - Number of threads and number of partitions do not need to match
    - Best performance if partitions are a multiple of threads (e.g., 12 vs. 4)
- Multi-threads enable aggregate summary reports

Subareas

- Microsimulator uses geographic subareas for multi-thread and MPI processing
  - Subarea numbers are defined in the network node file
    - SimSubareas assigns nodes to geographic subareas
  - Number of threads / slaves = number of subareas
    - “Extra” subareas are re-assigned to subarea zero
- Each subarea has its own simulation fidelity
  - None – no simulation is performed in the subarea (magic moves)
  - Macro – queues and saturation flow rates
  - Meso – cell-based simulation (~Version 4)
  - Micro – distance-based simulation (car following)

Input / Output Files

- All input and output file keys end with "_FILE"
- All output file keys start with "NEW_"
- Most data files include an optional "_FORMAT" key
  - Input files ignore the format key if a *.def file is found
  - The default file format is TAB_DELIMITED rather than VERSION3
- The value of PROJECT_DIRECTORY is added to the front of all file key values

PROJECT_DIRECTORY	c:/chicago
LINK_FILE	network/link.txt
Opens the link file:	c:/chicago/network/link.txt
Using the format found in:	c:/chicago/network/link.txt.def

Network Files

- Network directories have been dropped
  - NET_DIRECTORY and NEW_DIRECTORY
  - PROJECT_DIRECTORY is now used
- Network key names have changed
  - NET_LINK_TABLE → LINK_FILE
  - NEW_LINK_TABLE → NEW_LINK_FILE
- Data Services
  - Standard methods for reading and writing network files
  - Most Version 4 network files can be used without modification
    - Toll file needs to be converted/added to the Lane Use file
  - Data problems → warning messages rather than error messages

Basic Network Keys

- Version 4
  - NET_NODE_TABLE
  - NET_ZONE_TABLE
  - NET_SHAPE_TABLE
  - NET_LINK_TABLE
  - NET_POCKET_LANE_TABLE
  - NET_LANE_USE_TABLE
  - NET_TOLL_TABLE
  - NET_LANE_CONNECTIVITY_TABLE
  - NET_TURN_PROHIBITION_TABLE
  - NET_PARKING_TABLE
  - NET_ACTIVITY_LOCATION_TABLE
  - NET_PROCESS_LINK_TABLE

- Version 5
  - NODE_FILE
  - ZONE_FILE
  - SHAPE_FILE
  - LINK_FILE
  - POCKET_FILE
  - LANE_USE_FILE
  - LANE_USE_FILE
  - CONNECTION_FILE
  - TURN_PENALTY_FILE
  - PARKING_FILE
  - LOCATION_FILE
  - ACCESS_FILE

Traffic Controls and Transit Keys

▪ Version 4	▪ Version 5
– NET_UNSIGNALIZED_NODE_TABLE	– SIGN_FILE
– NET_SIGNALIZED_NODE_TABLE	– SIGNAL_FILE
– NET_PHASING_PLAN_TABLE	– PHASING_PLAN_FILE
– NET_TIMING_PLAN_TABLE	– TIMING_PLAN_FILE
– NET_DETECTOR_TABLE	– DETECTOR_FILE
– NET_SIGNAL_COORDINATOR_TABLE	– SIGNAL_FILE
– NET_TRANSIT_STOP_TABLE	– TRANSIT_STOP_FILE
– NET_TRANSIT_FARE_TABLE	– TRANSIT_FARE_FILE
– NET_TRANSIT_ROUTE_TABLE	– TRANSIT_ROUTE_FILE
– NET_TRANSIT_SCHEDULE_TABLE	– TRANSIT_SCHEDULE_FILE
– NET_TRANSIT_DRIVER_TABLE	– TRANSIT_DRIVER_FILE

Demand Files

- Trip and activity files consolidated into trip file
  - OD location/time + activity duration
- Household and person files combined
  - Vehicles numbered using household ID
- Household list → selection file
  - Household, person, tour, and trip selection options
- Additional vehicle type data
  - Operating cost and vertical grade impacts
- Skim files include OD size and time period meta-data
  - Partition time periods or merge time periods

Link Delay File

- Link Delay and Performance files are different, but interchangeable in most applications
  - Performance files include additional data fields
- Volume concept is replaced by flow rates
  - Version 4: volume = vehicles entering or exiting a link by time period
  - Version 5: flow = feet(meters) traveled in time period / link length
    - Travel units are Vehicles, Persons, or Passenger Car Equivalence
    - Converts VMT to vehicles that travel the full length of the link
    - Stored with one decimal place (e.g., 10.2 vehicles / 15 minutes)
- Travel times are similar
  - Travel time = link length * VHT / VMT during the time period

Plan Files

- All trip data is stored in a set of nested records
  - Eliminates leg coordination problems and simplifies processing
  - Significantly larger plan file with much more information
- The primary record includes
  - A full copy of the input trip file record
  - Path departure and arrival times
  - Trip travel time by mode (walk, drive, transit, wait, other)
  - Total trip length, cost, and impedance
- The nested records include
  - Mode, ID type, facility ID, travel time, distance, cost and impedance for each leg / link on the path

Version 4 Compatibility

- Version 5 automatically converts Version 4 network files on input (except for toll file)
- Many demand files reasonably convert automatically
  - Important exceptions include Plan, Population and Activity files
  - Vehicle, Trip and Vehicle Type files may have problems with vehicle types/subtypes and vehicle ID numbers
- NewFormat converts Version 4 files to Version 5
  - Since the files are read into memory before being written, large files should be processed separately