Intersection Delay Modeller - EMME Modeller

Naveen Juvva

AECOM

21st International
EMME Users' Conference

AECOM

Outline

- Background
- Why the Modeller Tool?
- Intersection Delays in EMME
- Intersection Delay Methodology
- Intersection Delay Calibration
- Construct of the Modeller Tool
- Travel Time Contour Results
- Observations and Conclusions

Background

- Travel time contours from the Regional model not realistically depicting existing conditions
  - Incorporating turn delays
- A post-assignment calculation of intersection turn delays
- Application of the intersection turn delays to feed back into the route choice
- Desired: Tool to run the above procedures

Why the Modeller Tool

- An independent utility that can be run in any setting
- Flexibility in terms of talking various components of EMME desktop and prompt console
- Modeller still provides connectivity to the macro platform, which can be called up if needed
- Python programming provides better organization of code compared to steps in macros
- A better troubleshooting support in the form of logbook

Intersection Delays in EMME

- Previous implementation of intersection delays in EMME
  - HCM Based turn volume delay functions: Terry Patridge
  - Intersection delay functions: Aashtiani and Iravani
- Options:
  - As a postprocessing tool to create network analysis
  - As part of the route choice

Intersection Delay Methodology

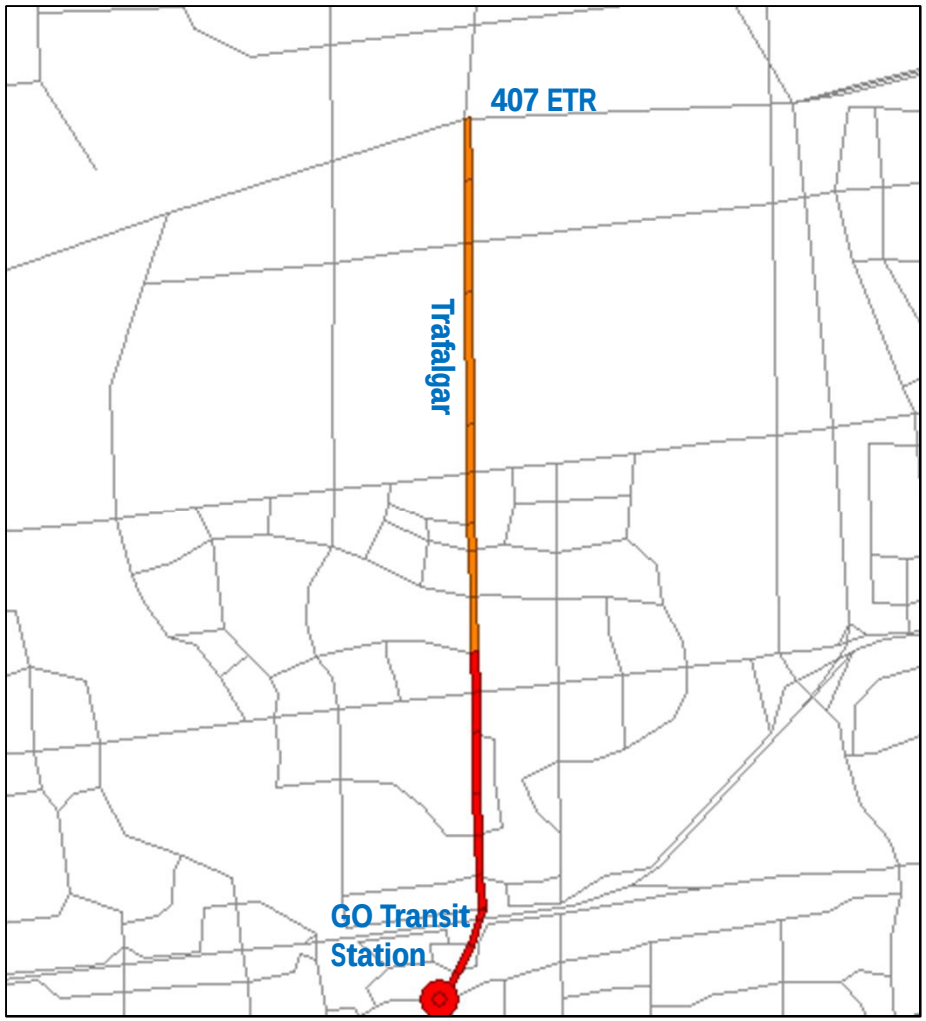
- Intersection control is input where available; elsewhere, a rule-based intersection control is applied
- Signals: Cycle length and splits are based on:
  - High – High
  - High – Low
  - Low – Low
- Unsignalized
  - Two-way stop-controlled
  - All-Way stop-controlled
- HCM Delay Equations with some simplified assumptions

Intersection Delay Calibration

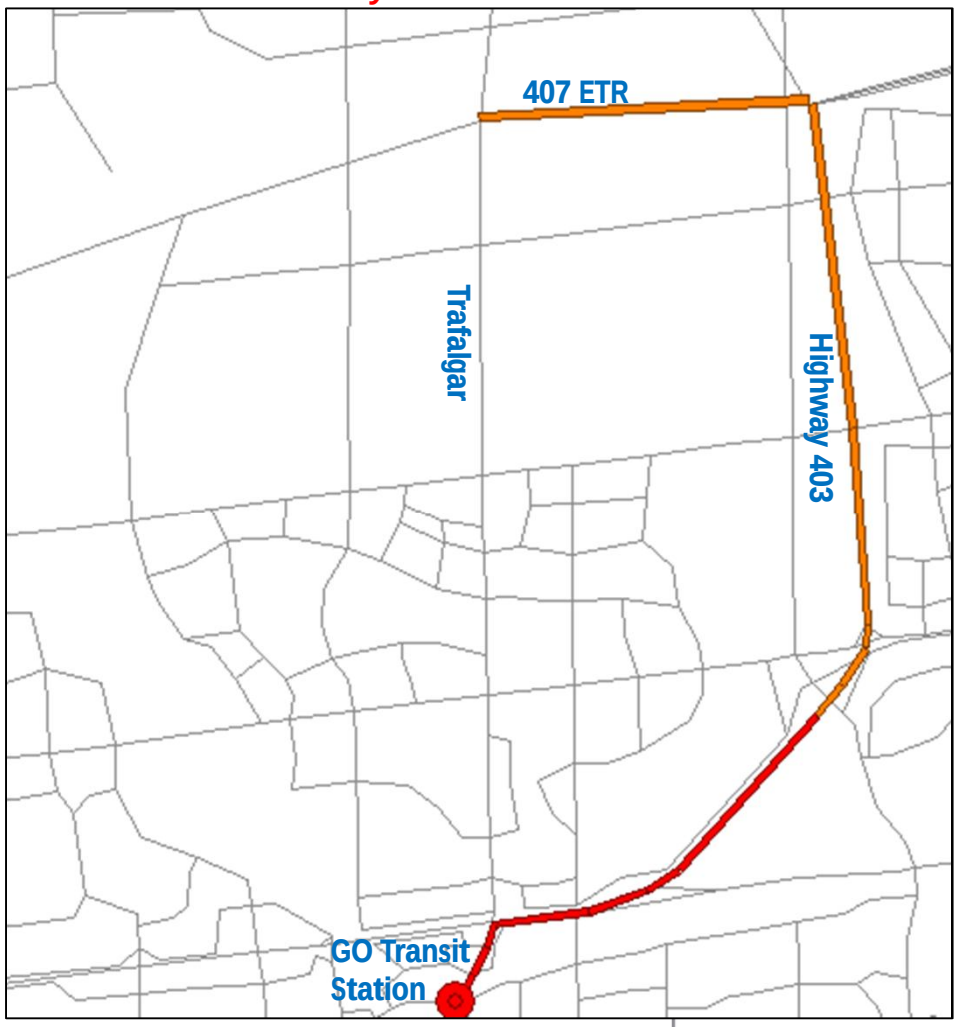
- Calibration of the intersection delay methodology
  - Corridor travel times: opportunity to calibrate signalized and unsignalized intersections
  - Resulting intersection turn delays and travel route patterns confirmed observations from the field

Intersection Delay Calibration

Shortest path with only link travel times



Shortest path with link travel times and intersection delays



AECOM

Intersection Delay Calibration

- Calibration of the intersection delay methodology
- Sample corridor travel time comparison

	Field travel times (min)	With only link delays (min)	With link and turn delays (min)
Southbound	14.6	7.8	13.9
Northbound	13.8	8.8	14.8

- Directionality was a problem with the model – independent of the intersection delays

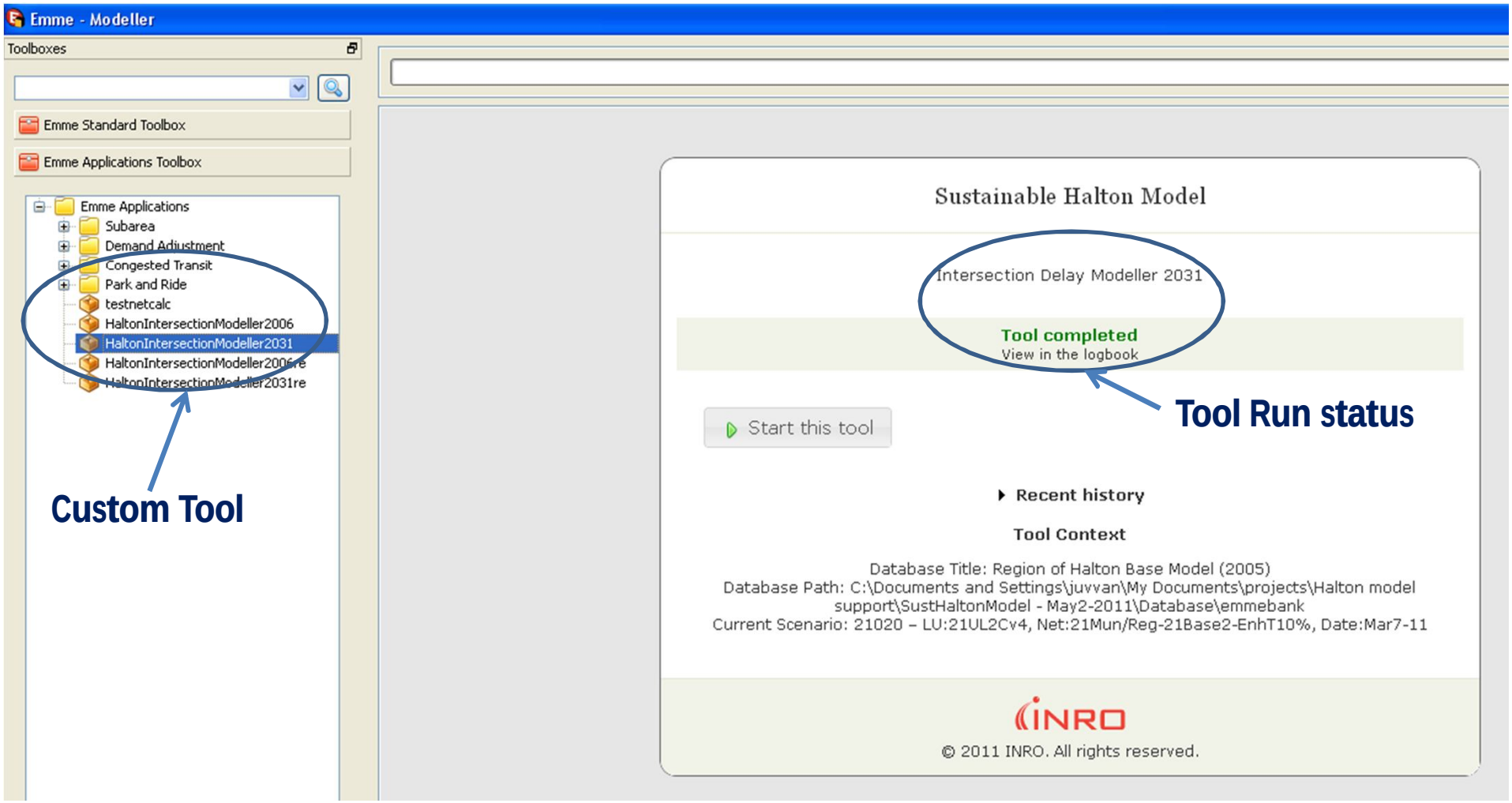
Construct of the Tool

- Import the python modules
  - eg: from __future__ import with_statement
- Define namespaces
  - eg: NETWORK_TOOL_NAMESPACE = "inro.emme.standard.data.network."
- Pagebuilder, etc
- Logbook trace
- Network, matrix and function editing procedures
- Run macros within to execute sub procedures
- Use desktop API reference: to access worksheets, etc

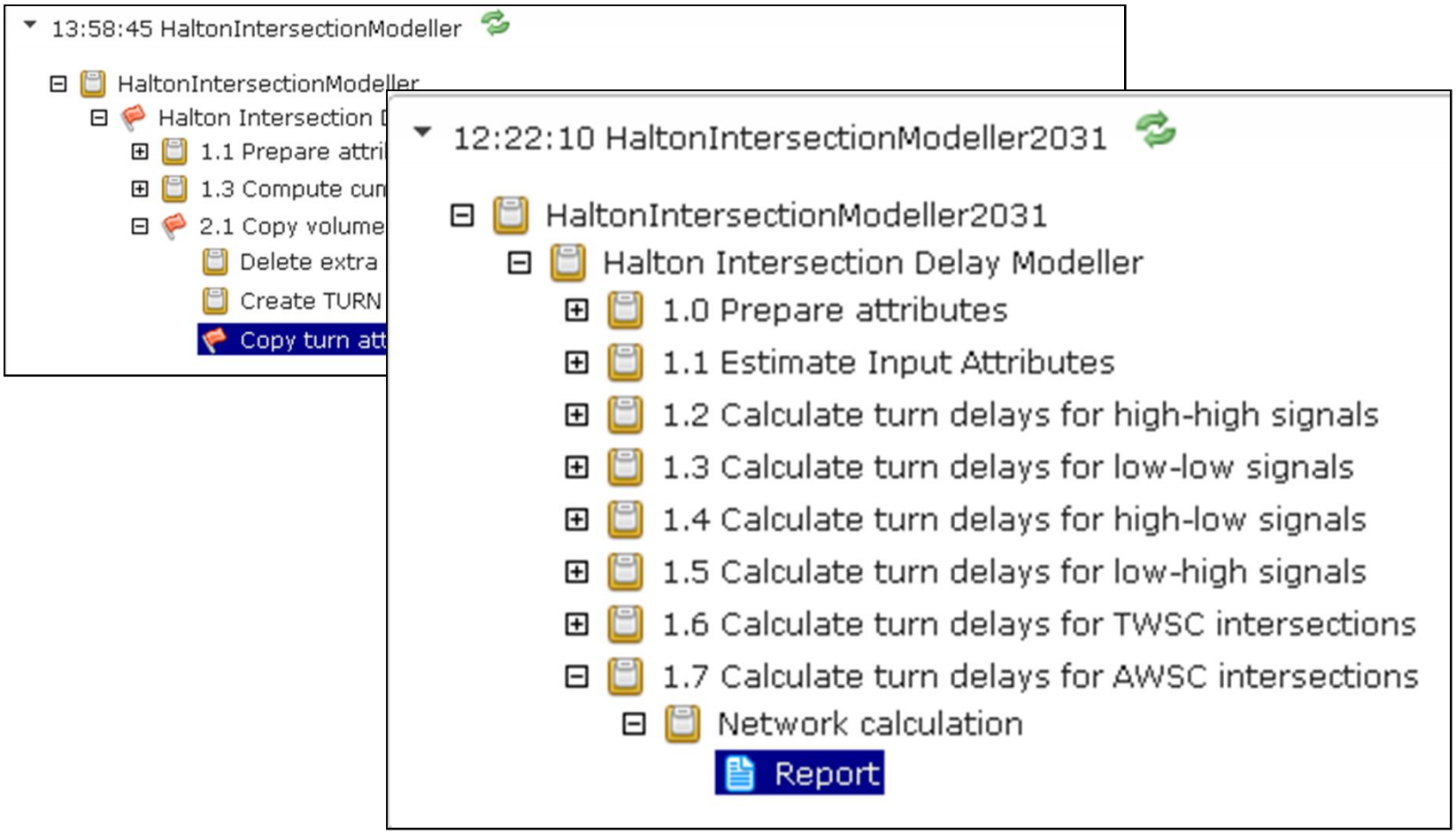
Construct of the Tool

```
with _m.logbook_trace(
        name="1.2 Calculate turn delays for high-high signals"):
        spec = {
          "type": "NETWORK_CALCULATION",
          "result": "@ptime",
          "expression": "(1/60)*(.5 * 120 * put(1 - put(15 / 120)) * get(2) / (1 - (1 .min. put( @pvol / 1700 * get(1))) * get(1)) \
                          + 900 * 1 * ((put(get(3) - 1))+ sqrt(get(4) * get(4) + 8 * .5 * get(3) / (1700 * get(1) * 1))))*(-105.le.@angle)*(@angle.le.-75)\
                         +(1/60)*(.5 * 120 * put(1 - put(60 / 120)) * get(2) / (1 - (1 .min. put(@pvol / 1900 * get(1))) * get(1))\
                         + 900 * 1 * ((put(get(3) - 1))+ sqrt(get(4) * get(4) + 8 * .5 * get(3) / (1900 * get(1) * 1))))*(-15.le.@angle)*(@angle.le.15)\
                        +.08*(75.le.@angle)*(@angle.le.105)",
          "selections": {
            "incoming_link": "@intj=1,2 and ul3=9999 or ul3=800,99999",
            "outgoing_link": "@int=1,2 and ul3=800,99999"
           }
         }
        compute_network(spec, scenario)
```

Implementation of the Tool



Implementation of the Tool

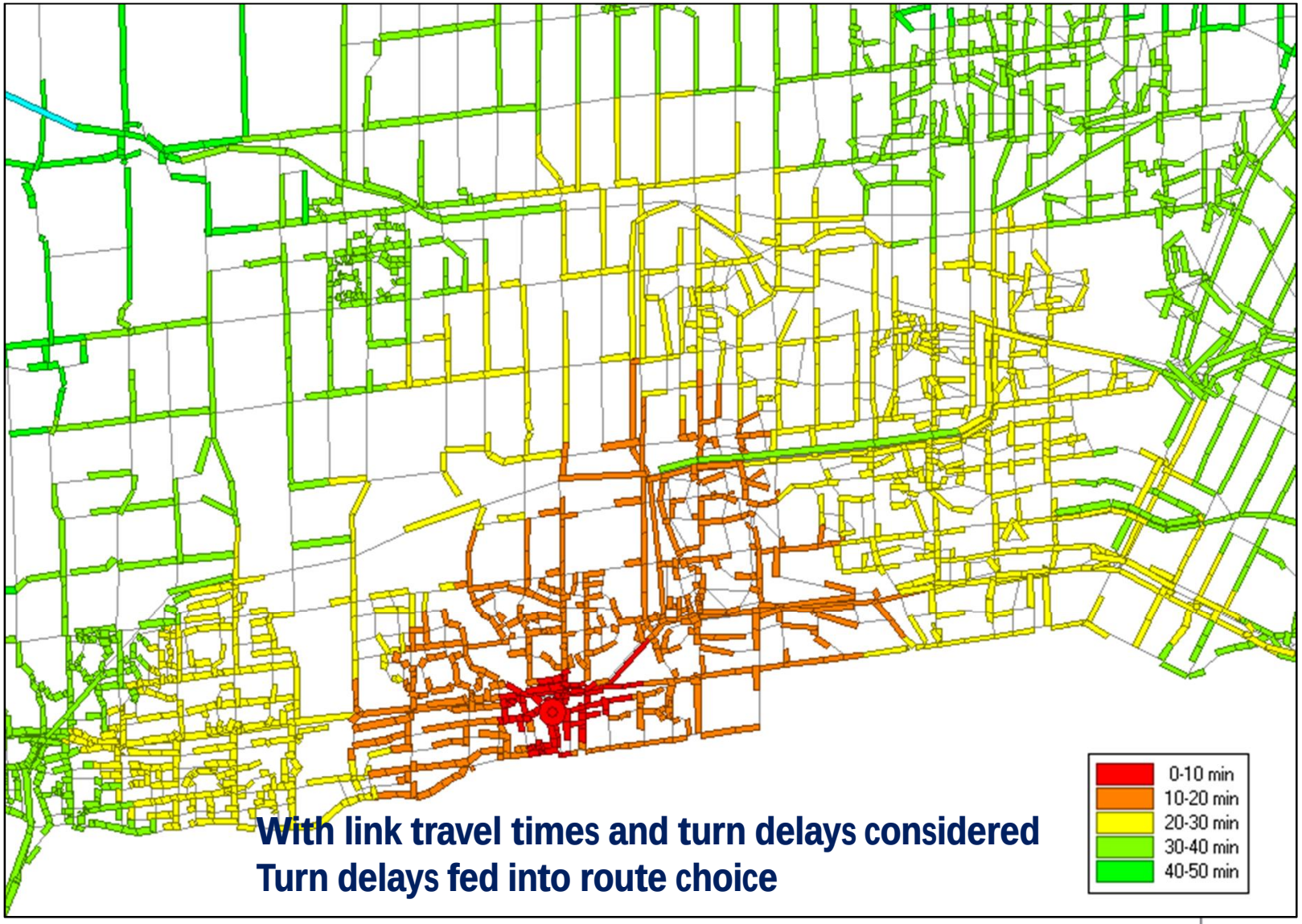


Travel Time Contours

Travel Time Contours

With link travel times and turn delays considered

Travel Time Contours



With link travel times and turn delays considered
Turn delays fed into route choice

Findings from Intersection Modeller

- Intersection turn delays reflect fairly closely the delays at signalized intersections
  - Protected left turns were not explicitly modelled
- Delays for minor approaches at TWSC and all approaches at AWSC are unreasonably high in some cases where the opposing turn volume (from oppvol.mac) are very high.
  - An upper limit was imposed
  - These are the cases where turn delays influenced route choice when they were incorporated into turn penalty functions

Not So Sweeping Conclusions

- Trouble shooting using logbook
  - The organized nature of the logbook trace is helpful to detect errors
- Run macros as part of the tool
  - Easier to call macros (at least for some users) in some cases
- Better organization and flexibility of code compared to macros – use of python scripting language medium
- Modeller is particularly helpful in creating utility tools that can be delivered to the client for them to incorporate into their toolbox
- Learning curve to create sophisticated tools or implement entire models in the Modeller
- Implementation of a particular travel demand model in the Modeller without use of other components of EMME