Flexible coupling of disaggregate travel demand models and network simulation packages ("IHOP2")
Final project report

Olivier Canella – WSP
Gunnar Flötteröd – KTH
Daniel Johnsson – KTH
Ida Kristoffersson – Sweco
Patryk Larek – WSP
Joacim Thelin – Sweco

CTS Working Paper 2016:3

Abstract
Trafikverket's development plan states as one of eleven expected results "En ny generation persontransportmodellsystem, med dynamisk modell för storstad implementerad" (Trafikverket, 2014).
IHOP aims to be this system. IHOP2 is the second development project advancing the IHOP system. IHOP2 couples the travel demand model Regent and the network assignment package TransModeler through a new, agent-based interface layer that is based on the MATSim transport simulation toolkit. The main objective of this effort is to demonstrate that such a coupling is feasible. This demonstration is delivered based on a prototypical Stockholm case study.
The present document is also meant to serve as a technical documentation of the IHOP system in its current form.
This project has been funded by Trafikverket under contract TRV2015/2950.

Keywords: strategic transport planning, dynamic traffic assignment, travel demand modeling, integrated transportation model system

Centre for Transport Studies
SE-100 44 Stockholm
Sweden
www.cts.kth.se

1 INTRODUCTION

IHOP2 continues the IHOP project (Almroth et al., 2014). IHOP investigated the possibility of replacing the static traffic assignment package Emme by a commercial dynamic traffic assignment (DTA) in Trafikverket's strategic transport forecasting model system. TransModeler was selected and was prototypically integrated with Regent, a micro-simulation based sketch version of Sampers.

A lesson learned from IHOP is that integrating different transport model components comes with substantial technical overhead. The objective of IHOP2 is to minimize this overhead by developing a modeling and implementation framework for the flexible and consistent integration of different travel demand models and network simulation packages (supply models). The software infrastructure deployed for this purpose is the multi-agent transport simulation toolkit MATSim (www.matsim.org). The result of this effort is subsequently referred to as "the IHOP system".

MATSim already comes with a set of model components that can be combined into different DTA configurations. On the demand side, this comprises models of location choice, mode choice, departure time choice and route choice. On the supply side, this comprises the simulation of vehicular traffic and public transport. MATSim is free software designed to be flexible and extendable, which renders it capable of interacting with external simulation components.

IHOP2 tackles technical, conceptual and experimental challenges. Technically, the implementation of interfaces between non-trivial software packages (Regent, TransModeler, MATSim) is required. Conceptually, the different modeling assumptions made in different packages require aggregation/disaggregation steps that enable these packages to exchange data. Experimentally, the plausibility of the resulting model system needs to be tested on a realistic Stockholm scenario.

The remainder of this report is structured as follows. Section 2 describes the overall design of the IHOP simulation system. Section 3 and 4 then detail the integration of respectively Regent and TransModeler in this system. Finally, Section 5 presents a Stockholm case study demonstrating the feasibility and plausibility of the IHOP model system.

2 SYSTEM DESIGN

The IHOP system decouples the involved (demand and supply) model components as much as possible. This reduces the overhead of adding new components and of replacing existing components, which simplifies the further development of the system. IHOP2 uses MATSim as an interface layer through which all other components are connected. This means that the insertion of a new demand or supply model component only requires implementing a single set of interfaces to the MATSim layer. This structure can be identified in Figure 1. The details of this figure will be explained later in this report.

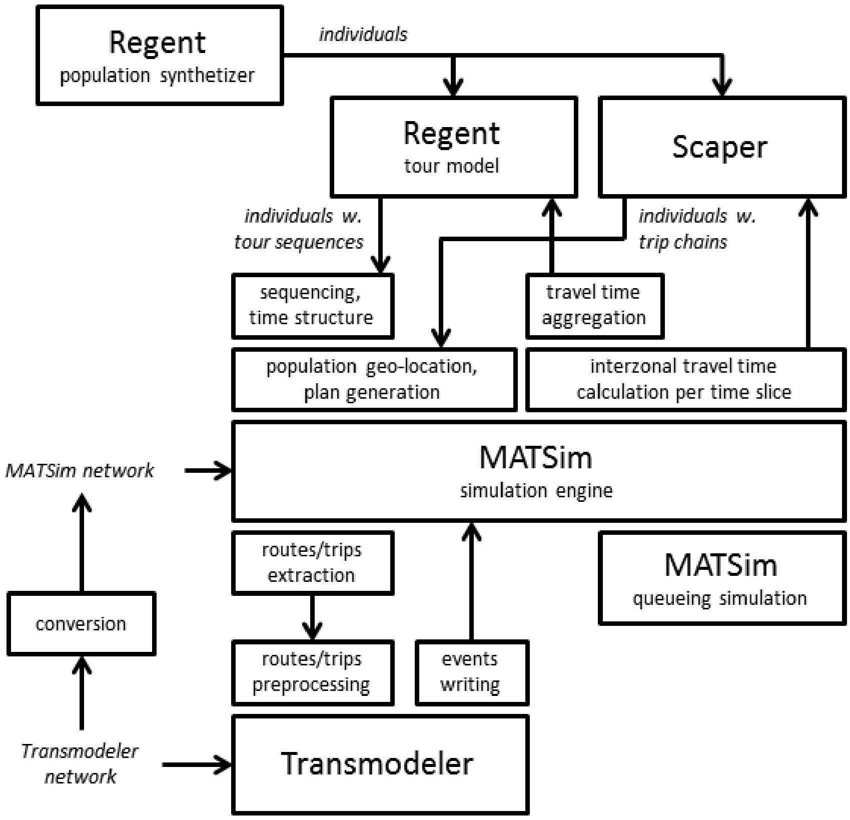

Figure 1: IHOP system structure

Regent is coded in C#, MATSim is coded in Java, and interaction with TransModeler is coded also in C# using TransModeler's API. A simple and robust way to let these programs interact is through file exchanges. For this purpose, a control program was written in Java that directly communicates with the MATSim interface layer through function calls. This allows to directly access the large amount of utility functionality already available in MATSim for agent-based simulation and analysis. The control program is parametrized through an XML file as exemplified in Figure 2. The control program calls Regent, TransModeler and MATSim in the right order, ensures that the data passed on between these components is in the right format, and computes various summary statistics.

```
<ihop2>

        <!-- GENERAL CONFIGURATION -->

        <!-- The shape file describing the zonal system. -->
        <zoneshapefile value="./input/sverige_TZ_EPSG3857.shp" />

        <!-- The shape file describing the buildings. -->
        <buildingshapefile value="./input/by_full_EPSG3857_2.shp" />

        <!-- The number of “outer” iterations between the demand model and MATSim. -->
        <iterations value="1" />

        <!-- MATSIM CONFIGURATION -->

        <!-- The MATSim configuration file. -->
```

```
        <matsimconfig value="./input/matsim-config.xml" />

        <!-- The fraction of the population created by the demand model that is
                to be simulated within MATSim. -->
        <matsimpopulationsubsample value="0.05" />

        <!-- TRAVEL DEMAND MODEL CONFIGURATION -->

        <!-- Left out, see Figure 4 -->

        <!-- NETWORK SUPPLY MODEL CONFIGURATION -->

        <!-- Left out, see Figure 10 -->

</ihop2>
```

Figure 2: Example of IHOP system configuration file

3 INTERACTIONS BETWEEN REGENT AND MATSIM

Regent takes as input a synthetic population (created by its own population synthesizer) and inter-zonal cost matrices. The matrices relevant to the IHOP system represent half-tour travel times, distances, and toll costs. Only the car mode is considered. Regent assigns to each synthetic individual a housing zone, optionally a zone for "work" and "other" activities, and defines by what mode the individual travels between these zones.

3.1 Regent/MATSim interface: concepts

To let Regent and MATSim interact, the population created by Regent needs to be inserted into MATSim, and the network performance measures simulated by MATSim need to be returned to Regent. To enable this interaction, two aggregation/disaggregation steps are implemented.
The first aggregation/disaggregation step accounts for the fact that Regent is static but MATSim is dynamic. The tour information attached to a Regent agent is hence enriched by a time structure. First, a sequence of the considered agent's tours is established; the current implementation only allows for the sequences home-work-home, home-other-home and home-work-home-other-home. Concrete departure times are then simulated by MATSim, which iteratively adjusts these times for each agent until an approximate equilibrium in terms of all day route/departure time choice is attained. This functionality is implemented in the "sequencing, time structure" component shown in Figure 1. Upon convergence, time-dependent inter-zonal travel cost matrices are computed by the corresponding component in Figure 1. In order to provide Regent with time-independent costs, an approximation of the expected value of these costs over the all-day time dimension is computed. This functionality is implemented in the "travel time aggregation" component shown in Figure 1 and follows the subsequently described logic.

1. Compute, for each departure time slot k, time-dependent shortest paths from all zones i to all zones j and memorize the corresponding inter-zonal travel cost $c(i, j, k)$.
2. Extract the following statistics:

- $f(x, k)$ is the estimated probability of departing in time slot k for a trip from *home* to $x \in \{work, other\}$.
- $g(x, \Delta k)$ is the estimated probability that activity $x \in \{work, other\}$ spans Δk time slots.

3. A *home*-based tour with *home* being located in zone i and purpose $x \in \{work, other\}$ being located in zone j then receives the tour cost

$$C(i, j, x) = \sum_{k} f(x, k) \left[c(i, j, k) + \sum_{\Delta k} g(x, \Delta k)\, c(j, i, k + \Delta k) \right].$$

This is an expected value of all tour costs over all possible departure time slots k, with each tour cost (in square brackets) being itself an expected value over all possible activity durations.

The second aggregation/disaggregation step accounts for the fact that the spatial resolution of Regent is at the level of traffic analysis zones, whereas MATSim does not use zones but real-valued coordinates. The zonal information delivered by Regent is hence disaggregated before being fed into MATSim. This is done by using supplementary building information. For each agent that is located in a zone, a building that matches that agent's activity type is sampled from that zone. (If no such building exists, a random point is sampled from within the zone geometry.) The nearest network link to that building is then chosen as the address of that agent's activity. This functionality is implemented in the "population geo-location, plan generation" component shown in Figure 1. The procedure is illustrated in Figure 3 (Larek, 2015).

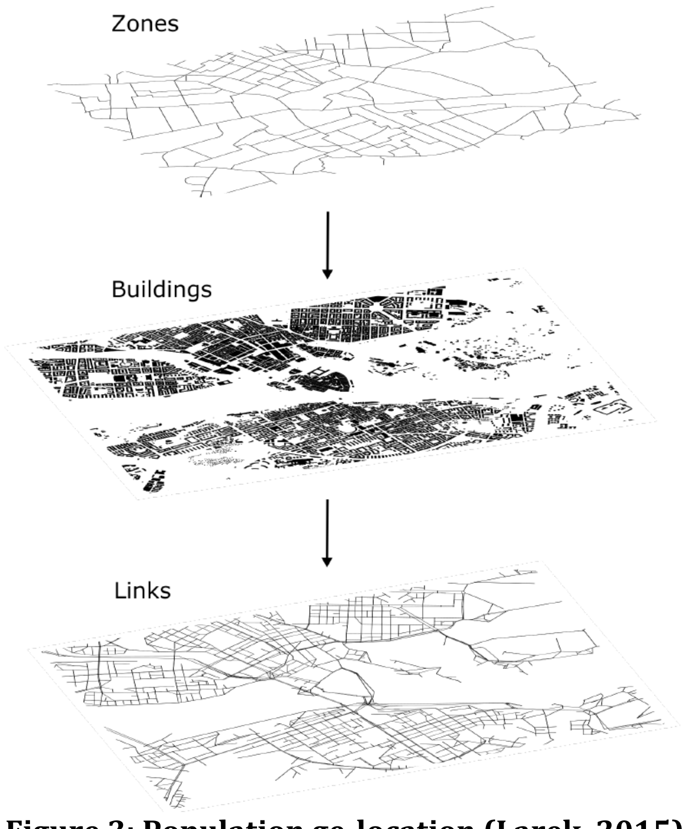

Figure 3: Population ge-location (Larek, 2015)

3.2 Regent/MATSim interface: technicalities

The interactions between Regent and MATSim are controlled by the following elements in the IHOP configuration file.

```
<ihop2>

        <!-- GENERAL CONFIGURATION -->

        <!-- Left out, see Figure 1 -->

        <!-- MATSIM CONFIGURATION -->

        <!-- The MATSim configuration file. -->
        <matsimconfig value="./input/matsim-config.xml" />

        <!-- The fraction of the population created by the demand model that is
                to be processed further for simulation. -->
        <matsimpopulationsubsample value="0.05" />

        <!-- DEMAND MODEL CONFIGURATION -->

        <!-- The used demand model. Possible values: regent, scaper. -->
        <demandmodel value="regent" />

        <!-- Location of the script that starts the demand model. -->
        <regentfolder value="./" />

        <!-- The command through which the demand model is called. -->
        <regentcommand value="cmd /c start /wait Regent.bat" />

        <!-- The fraction of the full population that is created by the demand model. -->
        <regentpopulationsample value="1.0" />

        <!-- Where the demand model is expected to write the agent information.
                Such a file is expected to exist at the beginning of the iterations, meaning
                that it is expected to exist BEFORE the demand model is called for the first
                time. -->
        <population value="./exchange/trips.xml" />

        <!-- Locations of the respective cost matrix files. -->
        <traveltimes value="./exchange/traveltimes.xml" />
        <distances value="./exchange/traveldistance.xml" />
        <tolls value="./exchange/traveltoll.xml" />

        <!-- The earliest time at which time-dependent travel costs are analyzed. -->
        <analysisstarttime value="04:00:00" />

        <!-- The time discretization of the dynamic travel costs. -->
        <analysisbinsize value="01:00:00" />

        <!-- The number of subsequent travel cost intervals that is being considered. -->
        <analysisbincount value="18" />

        <!-- The number of nodes sampled per zone when calculating inter-zonal travel
                times. -->
        <nodesamplesize value="1" />

        <!-- If roadpricing is to be accounted for. (This is a shortcut to simplify
                experiments; the roadpricing is normally set in the MATSim configuration
                file.) -->
        <usetoll value="false"/>
```

```
        <!-- SUPPLY MODEL CONFIGURATION -->

        <!-- Left out, see Figure 10 -->

</ihop2>
```

Figure 4: IHOP system configuration for interaction with Regent

Regent writes mode and destination for each tour of each individual into an XML file that complies with MATSim's ObjectAttribute specification (http://matsim.org/files/dtd/objectattributes_v1.dtd) and is indicated in the configuration file throug the `population` element. Here is an example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE objectAttributes SYSTEM "http://matsim.org/files/dtd/objectattributes_v1.dtd">
<objectAttributes>
    <object id="1">
        <attribute name="homezone" class="java.lang.String">123</attribute>
        <attribute name="workzone" class="java.lang.String">456</attribute>
        <attribute name="otherzone" class="java.lang.String">789</attribute>
        <attribute name="worktourmode" class="java.lang.String">car</attribute>
        <attribute name="othertourmode" class="java.lang.String">pt</attribute>
        <attribute name=”birthyear” class=”java.lang.Integer”>1976</attribute>
        <attribute name=”sex” class=”java.lang.String”>m</attribute>
        <attribute name=”housingtype” class=”java.lang.String”>villa</attribute>
        <attribute name=”income” class=”java.lang.Integer”>30000</attribute>
    </object>
</objectAttributes>
```

Figure 5: Regent population file example

The following table enumerates all currently considered attributes. Non-required attributes may be left out in the XML file. All variables are case-sensitive.

attribute name	**required**	**type**	**possible values**
homezone	yes	java.lang.String	Any valid zone identifier.
workzone	no		
otherzone	no		
worktourmode	yes if workzone is defined	java.lang.String	Car, None, CarPassenger, PublicTransport, Bicycle, Walk
othertourmode	yes if otherzone is defined		
birthyear	no	java.lang.Integer	A non-negative integer number.
sex	no	java.lang.String	“m” or “f”
housingtype	no	java.lang.String	“apartment”, “villa”

			(If omitted then a random point within the zone is sampled.)
income	no	java.lang.Integer	A non-negative integer number.
car	no	java.lang.Boolean	“true”, “false”

Figure 6: Regent population file entries

MATSim writes inter-zonal cost matrices in an existing MATSim matrix file format as specified in http://matsim.org/files/dtd/matrices_v1.dtd. One file is written per cost type (travel time, distance, monetary cost); the corresponding file names are indicated in the `traveltimes`, `distances`, `tolls` elements in the configuration file. Time is expressed in minutes, distance in kilometers and toll in SEK. All costs are half-tour costs (meaning that the tour travel time is divided by two, which is consistent with the way in which EMME travel times are processed by Regent). Here is an example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE matrices SYSTEM "http://matsim.org/files/dtd/matrices_v1.dtd">
<matrices>
<!-- ====================================================================== -->
              <matrix id="OTHER" desc="other tour costs, averaged over entire population">
                            <entry from_id="fromId1" to_id="toId2" value="11.0" />
                            <entry from_id="fromId1" to_id="toId3" value="22.0" />
                            <entry from_id="fromId4" to_id="toId4" value="33.0" />
              </matrix>
<!-- ====================================================================== -->
              <matrix id="WORK" desc="work tour costs, averaged over entire population">
                            <entry from_id="fromId1" to_id="toId2" value="12.34" />
                            <entry from_id="fromId1" to_id="toId3" value="56.78" />
                            <entry from_id="fromId4" to_id="toId4" value="90.12" />
              </matrix>
<!-- ====================================================================== -->
</matrices>
```

Figure 7: MATSim cost matrix file example

4 INTERACTIONS BETWEEN TRANSMODELER AND MATSIM

4.1 Transmodeler → MATSim network transformation

A program that transforms a TransModeler network into a MATSim network was developed. Relying on a TransModeler network instead of creating a MATSim network in some other way, for instance from readily available OpenStreetMap data, has the advantage that only one network representation needs to be manually maintained. Also, the necessary consistency between the network references in MATSim’s travel plans and TransModeler’s trips is automatically ensured.

The TransModeler network is represented by five relevant files in CSV-Format. In the following, the relevant entries in each of these files are explained.

The **TransModeler nodes file** contains all data necessary to define the MATSim nodes. Specifically:

TransModeler node attribute	**MATSim node attribute**	**Comment**
ID	id	
Longitude	x	TM coordinates are multiplied by 10^6 and then transformed from WGS84 to WGS84_SWEREF99
Latitude	y	

Table 1: Relevant entries in TransModeler node file

The **TransModeler links file** contains partial data necessary to define the MATSim links:

TM link attribute	**MATSim link attribute**	**Comment**
ID, AB	id (for link in direction AB)	A bidirectional TM link may have ID="123", AB="NW" and BA="SE". This is turned into the MATSim links "123_NW" and "123_SE".
ID, BA	id (for link in direction BA)	
ANode	from (for link in direction AB) to (for link in direction BA)	The above mentioned example link may have ANode="10" and BNode="20". MATSim link "123_NW" then goes from "10" to "20", and MATSim link "123_SE" goes from "20" to "10".
BNode	from (for link in direction BA) to (for link in direction AB)	
Class	freespeed, capacity (in both directions)	See details in Table 3.

Table 2: Relevant entries in TransModeler link file

The TransModeler road classes are mapped to MATSim link parameters as follows:

Link type	**capacity [veh/h]**	**free speed [km/h]**
Undefined	1200	56
Freeway	2400	113

Rural Highway	2000	97
Expressway	2200	97
Ramp	1800	72
System Ramp	2000	80
Minor Arterial	1000	72
Major Arterial	1200	80
Minor Collector	800	56
Major Collector	900	56
Local Street	600	48
Access Road	600	48
Trail or Other Local Road	500	40
Roundabout	600	40
Tunnel	1200	80
Rail	600	121
Waterway	600	72
SV70_E	1800	72
SV70	2000	72

Table 3: Road types in TransModeler link file

The **TransModeler segments file** contains the remaining data necessary to define the MATSim links:

TM segment attribute	MATSim link attribute	Comment
Link, AB Link, BA	id	See Table 2.
Position	--	Multiple segments per link are distinguished by their position; MATSim does not distinguish segments.
Length, Lanes_AB Length, Lanes_BA	permlanes	Permlanes (number of lanes in MATSim) is computed as length-weighted average of

		segment lanes.
Length	length	MATSim length is sum of TM segment lengths.

Table 4: Relevant entries in TransModeler segment file

Additional connectivity information that extracted from the TransModeler lanes and lane connectors file is stored in an additional MATSim lane definition file. The resulting MATSim network needs to be post-processed in order to ensure that every link can be reached from every other link. Otherwise, problems with MATSim's routing mechanisms arise. Only the largest connected component of the MATSim network is hence maintained. Figure 8 gives a visual impression of the effect of this postprocessing step.

Raw network	**Postprocessed network**
	
	

Figure 8: Effect of network postprocessing[1]

One may assume that the removal of links in the network postprocessing leads overall to an under-estimation of network capacity. This effect is yet to be quantified in detail; a simple counter-measure taken within MATSim is to scale up the the space and flow capacities of the remaining links.

4.2 TransModeler/MATSim interface: concepts

[1] Created with the VIA software (http://www.senozon.com/products/via).

Given a list of trips (consisting, essentially, of a departure node and time, a route, and a destination node), TransModeler executes these trips and by default returns a variety of network performance measures (Caliper, 2016). Agents in MATSim, on the other hand, do not choose between individual trips but between all-day travel plans (comprising all-day trip chains), the performance of which is evaluated based on the individual-level experience of each agent. This experience is represented by a stream of events that describe everything the agent experiences in the network.

In order to insert MATSim agents into TransModeler, their all-day travel plans are split into a set of trips, which are then written out in TransModeler's proprietary file format. TransModeler then moves one vehicle for each trip through the network. Using TransModeler's proprietary scripting language, an event stream is written directly in MATSim's event file format as the network simulation runs. Once the network simulation is complete, MATSim reads these events and, amongst other things, computes the score (utility) of the executed plan of each agent. Based on this information, the agents then select new plans to be executed in the next iteration. This process is iterated until an approximate stochastic equilibrium is identified based on the performance measures computed in MATSim. Two major challenges had to be tackled in order to realize these interactions.

A first difficulty was related to the fact that TransModeler uses lanes and accounts for turning move restrictions, whereas MATSim (at least in its default configuration) does not. An optional MATSim facility (the "signals contrib") hence had to be invoked to ensure that the MATSim router would comply with TransModeler's turning move restrictions. However, it appeared unavoidable that TransModeler changed certain paths defined by MATSim; the reason for this is unclear. This in turn had the effect that the event stream returned by TransModeler included references to network links that were not contained in the post-processed MATSim network, which had the effect that the MATSim event handling mechanism terminated with an error message. The following cure to this problem was hence implemented on the MATSim side: When mapping the synthetic population produced by Regent onto the network, it was ensured that all activities take place on links within the largest connected network component. This ensures that at least one path exists between all activity locations. The actual network used in the simulation was, however, not post-processed, meaning that it contained links that were not reachable by other links. The result of this is that (i) the MATSim router is always able to connect activity locations with a feasible route and that (ii) the MATSim event handling mechanism knows all links that also exist in the TransModeler network, which robustifies it against unwanted route-changes within TransModeler.

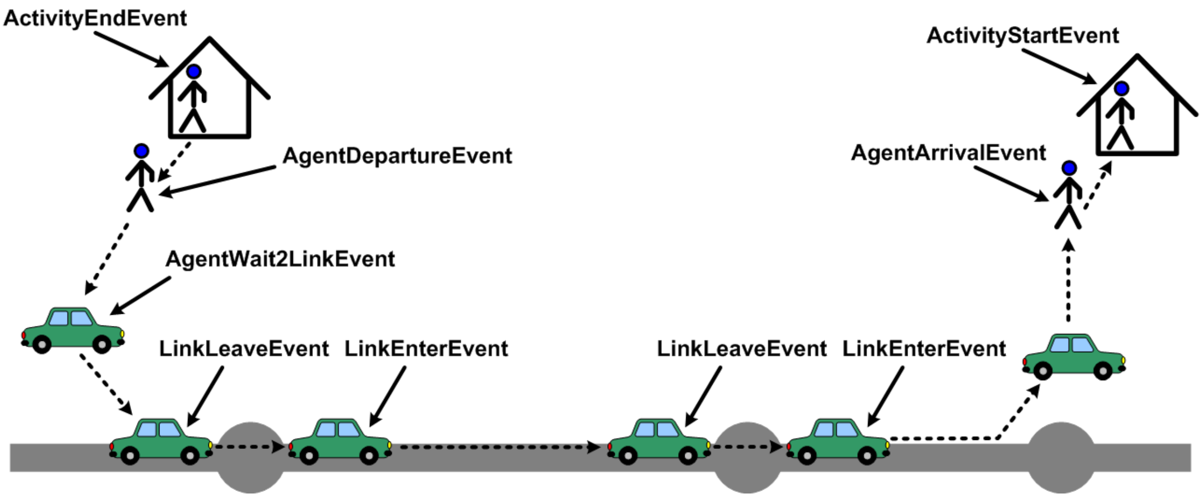

Figure 9: Sequence of events as expected by MATSim

A second difficulty resulted from the fact that TransModeler executes trips, whereas MATSim executes trip chains. In MATSim, an agent is only able to leave a location after it has arrived there; meaning that the start time of trip *n*+1 is never before the end time of trip *n* in the same trip chain. When feeding a trip chain into TransModeler, however, all trips are executed by *independent* vehicles. If it now happens that trip *n* is not completed before trip *n*+1 is started, then TransModeler returns to MATSim for the corresponding agent a "departure" event (start of trip *n*+1) that takes place before the "arrival" event indicating that trip *n* has been completed. This leads to the premature termination of the MATSim event handling mechanism in an error state, and it is rather likely to happen in early iterations where there is severe congestion during the not-yet-equilibrated morning rush hour. This problem was tackled jointly with the related but less severe issue that events obtained from TransModeler are not correctly ordered within a same second. Indeed time stamps are given per 0.1 second but event stream within a same second are collected by TransModeler only at the end of the specific second. The event stream received by MATSim is hence pre-processed by a component that (i) corrects all timestamps such that their chronological ordering is maintained and (ii) is capable of rearranging agent-specific events such that their logical (trip-chain) ordering is maintained.

4.3 TransModeler/MATSim interface: technicalities

The interactions between TransModeler and MATSim are controlled by the following elements in the IHOP configuration file.

```
<ihop2>

        <!-- GENERAL CONFIGURATION -->

        <!-- Left out, see Figure 1 -->

        <!-- MATSIM CONFIGURATION -->

        <!-- Left out, see Figure 1 -->
```

```
        <!-- DEMAND MODEL CONFIGURATION -->

        <!-- Left out, see Figure 2 -->

    <!-- SUPPLY MODEL CONFIGURATION →

        <!-- Possible values: TransModeler, matsim -->
        <mobilitysimulation value="TransModeler" />

        <!-- TransModeler-specific: Where additional link attributes are stored. -->
        <linkattributefile value = "./input/link-attributes.xml"/>

        <!-- TransModeler-specific: Where lane information is stored. -->
        <lanesfile value = "./input/lanes20.xml"/>

        <!-- TransModeler-specific: The paths file. -->
        <paths value="./exchange/paths.csv"/>

        <!-- TransModeler-specific: The trips file. -->
        <trips value="./exchange/trips.csv"/>

        <!-- TransModeler-specific: The events file. -->
        <events value="./exchange/events.xml" />

        <!-- TransModeler-specific: Location of the script that starts TransModeler. -->
             <TransModelerfolder value="./tsm" />

        <!-- TransModeler-specific: The command through which TransModeler is called. -->
        <TransModelercommand value="cmd /c start /wait tsm_matsim.exe" />

</ihop2>
```

Figure 10: IHOP configuration file for interaction with TransModeler

MATSim's trip chains are split into separate trips, which are written into two files comprehensible to TransModeler: a paths file (indicated by the `paths` element in the configuration file) and a trips file (indicated by the `trips` element in the configuration file).
The paths file contains all routes (link sequences) used by all vehicles in the given iteration; an example is shown in Figure 11 for the four first routes. (Positive/negative link identifiers indicate the direction in which a link is traversed.)

```
1 { 95293 50024 -88934 -116494 -1636 -37471 -110173 -99347 -108261 -22901 -45008 -79993 -
27937 -48491 -69579 -116073 -30343 -54638 -54247 -75382 -9089 -11820 -11821 -80393 -35443 -
12971 -79987 -115665 -116501 65696 -41434 -90160 89314 -99349 -48891 -64121 -88942 -76068 -
100464 75351 125359 51928 51929 92866 26388 69146 85772 32317 25 41392 67982 62105 18138
116882 18962 -86666 -25669 -36405 -36404 -35504 9160 46243 90228 87382 114602 114603 57809
87748 87749 47427 87385 111799 -8007 -74288 -100932 -91386 }
2 { -102035 -76160 -87009 36404 36405 25669 86666 37980 -116882 -18138 109810 1253 6289
60895 -59367 105880 80770 62098 78432 99290 124790 77626 39776 125722 96861 100465 64121
48891 99349 7937 109374 40996 40995 87344 43442 7918 97299 76892 }
3 { 76892 95293 50024 -88934 -116494 -1636 -37471 -110173 -99347 -108261 -22901 -45008 -
79993 -27937 104771 110594 -24461 -4374 -29130 -44191 -22905 -24462 -48490 -28059 -35881
109373 -104770 29129 63715 23709 38615 35884 101289 71112 90906 35885 97677 -92906 -67241 -
67240 5184 114905 114906 88184 111779 56257 -10251 101302 -63759 99396 6353 50041 50042 -
84999 }
4 { 5998 -99396 -17359 -99395 -72674 -72258 51206 82352 -56618 -91310 -25623 71114 60943 -
4353 -11503 95684 97299 }
```

Figure 11: Example paths file (paths 1 to 4).

The trips file contains all trips made in the given iteration, one trip per row. An example of this file is shown in Figure 12. The comma-separated columns of this file contain the following information:

1. the trip identifier,
2. the origin location identifier,
3. the destination location identifier,
4. the type of the origin location (a node in our case),
5. the type of the destination location (a node in our case),
6. the vehicle class (car, truck …),
7. the origin link,
8. the path identifier (referring to an entry in the paths file, Figure 11),
9. the destination link,
10. the departure time of the trip (in seconds after midnight),
11. the type of the last activity performed before the trip was started,
12. the type of the activity to be performed at the destination of the trip,
13. the identifier of the agent executing the trip.

The last three items (activity types and agent identifier) are not needed by TransModeler to execute the trip but provide information that is required in the event stream being returned by TransModeler to MATSim.

```
ID,OriID,DesID,OriType,DesType,Class,OriLink,Path,EndLink,DepTime,OriAct,EndAct,AgentID
4987,14623,10750,Node,Node,PC1,-80424,4969,7028,21601.0,home,work,1418820
9521,7509,208,Node,Node,PC1,24811,9463,121585,21601.0,home,work,1781314
1817,7941,14806,Node,Node,PC1,-63774,1813,-86092,21603.0,home,work,1149061
16311,7583,11398,Node,Node,PC1,26219,16187,16250,21603.0,home,work,612055
20379,2304,6619,Node,Node,PC1,-74214,20228,60881,21603.0,home,work,996932
17561,10248,10821,Node,Node,PC1,-51514,17427,-7920,21604.0,home,other,689279
```

Figure 12: Example trips file (trips 1 to 6).

The events file written by TransModeler follows a MATSim XML file format (Zilske, forthcoming) and represent the logic illustrated in Figure 9. An example is shown in Figure 13. One identifies the aforementioned activity types (`actType` elements) and person identifiers (`person` elements).

```
  <?xml version="1.0" encoding="utf-8" ?>
- <events version="1.0">
    <event actType="home" link="80425_SW" person="1424972" type="actend" time="21601" />
    <event link="80425_SW" person="1424972" type="departure" time="21601" legMode="car" />
    <event person="1424972" type="PersonEntersVehicle" time="21601" vehicle="1424972" />
    <event link="80425_SW" person="1424972" type="wait2link" time="21601.0" vehicle="1424972" />
    <event actType="home" link="104616_W" person="1649108" type="actend" time="21601" />
    <event link="104616_W" person="1649108" type="departure" time="21601" legMode="car" />
    <event person="1649108" type="PersonEntersVehicle" time="21601" vehicle="1649108" />
    <event link="104616_W" person="1649108" type="wait2link" time="21601.0" vehicle="1649108" />
    <event actType="home" link="105150_NE" person="1321040" type="actend" time="21601" />
    <event link="105150_NE" person="1321040" type="departure" time="21601" legMode="car" />
    <event person="1321040" type="PersonEntersVehicle" time="21601" vehicle="1321040" />
    <event link="105150_NE" person="1321040" type="wait2link" time="21601.0" vehicle="1321040" />
    <event actType="home" link="38664_S" person="1405326" type="actend" time="21603" />
    <event link="38664_S" person="1405326" type="departure" time="21603" legMode="car" />
    <event person="1405326" type="PersonEntersVehicle" time="21603" vehicle="1405326" />
    <event link="38664_S" person="1405326" type="wait2link" time="21603.0" vehicle="1405326" />
    <event link="80425_SW" person="1424972" type="left link" time="21602.4" vehicle="1424972" />
    <event link="118493_S" person="1424972" type="entered link" time="21602.4" vehicle="1424972" />
    <event actType="home" link="111192_E" person="1631026" type="actend" time="21604" />
    <event link="111192_E" person="1631026" type="departure" time="21604" legMode="car" />
```


Figure 13: Example events file.

5 STOCKHOLM CASE STUDY

The case study focuses on the Greater Stockholm region. Its purpose is

- to demonstrate the practical and computational feasibility of using the IHOP2 system in strategic planning, and
- to exemplify the added value of of moving from a static and aggregate to a dynamic and disaggregate representation.

The purpose of the case study is not to deliver a fully developed and calibrated model system; plausible model system outputs and reactions to changes in input parameters are considered as sufficient indication of the functionality of the current prototype.

5.1 Description of the Stockholm case study and realization plan

Figure 14 shows the (preprocessed) road network of the study region. The network consists of 11'974 nodes and 22'547 links. Each of the 2’642 traffic analysis zones that contains at least one network node is included in the case study, in the sense that all tours that start and end in such a zone are simulated.

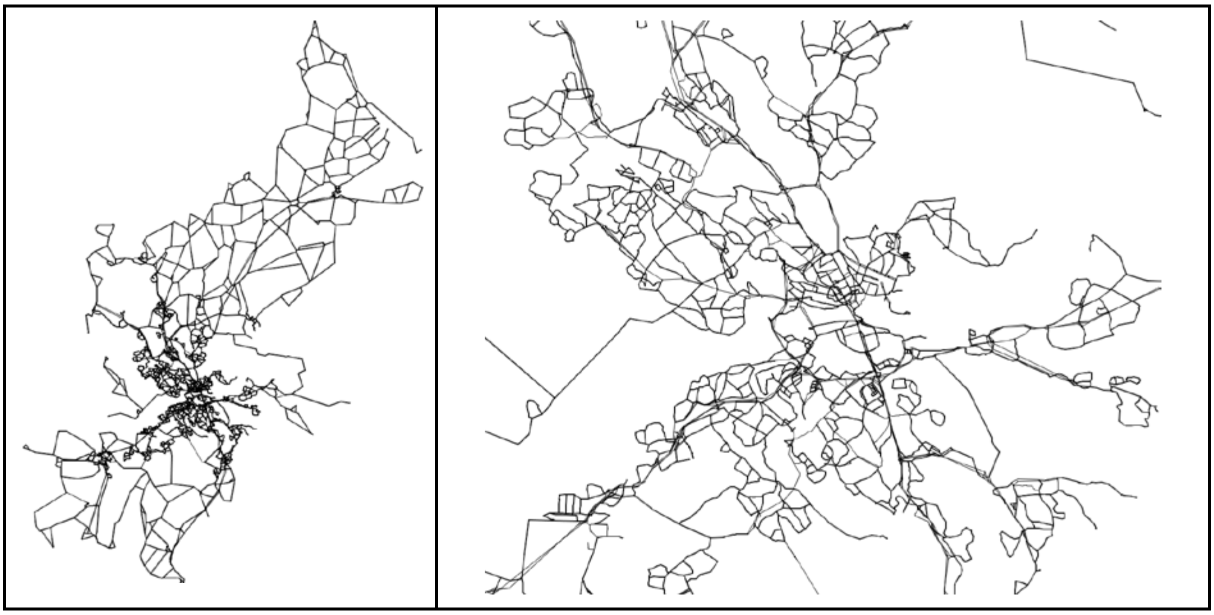

Figure 14: Study region network[2]

The data sets used in the Stockholm case study are enumerated and described in Table 5.

Data set	**Source and explanation**
Synthetic population.	Created by the Regent population synthesizer prior to the simulation experiments.
Typical activity durations and facility opening times	Estimated based on common sense. There was no point in extracting exact data from survey data given the uncalibrated overall system. See Table 6 for details.
Utility function parameters	Taken from an existing Contram model for Stockholm. See Table 6 for details.
Shape files of traffic zones and building geometries.	Retrieved from maps.slu.se.
Road network.	A TransModeler network was created at WSP from NVDB (nationell vägdatabas) during IHOP project. A corresponding MATSim network was automatically extracted from the TransModeler network.
Toll station locations and toll profile.	The locations were encoded in the TransModeler network. The toll profile is broadly known. See also Table 8.

[2] Created with the VIA software (http://www.senozon.com/products/via).

Table 5: Data used in the Stockholm case study

Simulation experiments of this case study required to access the three different programs Regent, TransModeler and MATSim (the latter in conjunction with the control program for the entire IHOP system). Access and usage skills for each program were distributed roughly as follows over the project participants: Regent at Sweco with support from KTH, TransModeler at WSP, MATSim at KTH. All simulations involving Regent were hence run at Sweco and all simulations involving TransModeler were run at WSP.

This separation of simulation efforts was enabled by the IHOP system design. As explained before and shown in Figure 1, Regent and TransModeler interact exclusively through the MATSim interface layer. This means that being able to interface Regent and MATSim and, separately from that, being able to interface TransModeler and MATSim is sufficient to demonstrate the technical compatibility of the entire system.

The Regent/MATSim and TransModeler/MATSim simulation experiments were hence run independently on different computers but were based on using identical MATSim interface layers and configurations. Since already these experiments pushed the involved computers to their (random access memory) limits, the joint execution of all programs would have required a stronger workstation than what was available at the time of project implementation. (A workstation with two processors (16 cores each) and 128 RAM has been purchased in the beginning of 2016 by KTH, meaning that future experiments are less likely to experience these computational constraints.)

5.2 Simulation studies with Regent and MATSim

Two large simulation studies were conducted, one without and one with Stockholm's time-dependent road pricing included in the model system. Both simulations comprised three (outer) iterations between Regent and MATSim, with MATSim running for 400 iterations in each outer iteration until an approximate route/departure time equilibrium was postulated. The entire experiment took about 45 hours on a laptop computer with Intel Xeon E645 processor, 2.4 GHz clock rate and 12 GB memory.

Regent simulated in each iteration the trip-making, location choice and mode choice of the full case study population. Five percent of this population were then sampled and fed into MATSim. Realistic network congestion patterns were obtained in MATSim by scaling the space and flow capacities of all links in the road network accordingly. This is possible because of the relative simplicity of MATSim's mesoscopic built-in traffic flow simulation; the validity of this approach in terms of traffic flow theory has been shown elsewhere (Flötteröd, forthcoming). MATSim then equilibrated route and departure time choice using the parameters shown in Table 6. The underlying utility function is detailed by Nagel et al. (forthcoming).

Parameter	**Value**
activity duration coefficient	60 h^{-1}
travel distance coefficient	-0.5 km^{-1}

marginal utility of money	1.0 SEK^{-1}
typical "home" duration	14:30
typical "work" duration	08:00
typical "other" duration	01:30
"home" opening times	00:00 – 24:00
"work" opening times	07:30 – 17:30
"other" opening times	07:30 – 21:30

Table 6: MATSim model parameters

The detailed computing times for this configuration and a measure of convergence are displayed in Table 7. The first column in this table indicates the outer (between Regent and MATSim) iteration number, and the second and third column the runtimes of MATSim and Regent, respectively. (The remaining computing time was consumed by the control program and interface operations such as the sampling of activity locations and the computation of cost matrices.) The last column displays the population logsum evaluated in Regent at the end of each outer iteration; a stabilization of this value indicates (but does not prove) convergence of the system as a whole. The first data row displays only the logsum resulting from a preparatory run of Regent with the network model Emme.

Outer iteration	**MATSim runtime [hh:ss]**	**Regent runtime [hh:ss]**	**Logsum [10^4]**
			2428
1	12:22	00:27	2054
2	14:49	00:27	2048
3	13:45	00:28	2050

Table 7: Regent/MATSim runtimes and convergence

Little efforts were made to tune the computational performance of the involved programs with respect to the needs of the IHOP system. For instance, each MATSim simulation was started from an uncongested network, even though it would be possible to initialize it with the converged travel times of the previous outer iteration. Another example is that Regent always simulates a full population, even though MATSim only simulates a fraction of that population. This suggests that the indicated performance measures are pessimistic estimates of what a fully developed system would be able to deliver.

The logsums change rather dramatically after the first iteration but then settle very quickly. The initial strong change indicates substantial differences between the Regent/MATSim and the Regent/EMME system. The further interpretation of this effect is, however, difficult because the IHOP system is completely uncalibrated. A possible explanation for the quick stabilization of the logsums after the first iteration is that MATSim allows for time choice, whereas EMME does not: This gives the simulated travelers an additional degree of freedom along the time dimension when trying to exploit the available network capacity, suggesting that this capacity is better utilized when time choice is available. This in turn may lead to a more damped response of network travel times to changes in demand levels and patterns.

Figure 15 illustrates the convergence of MATSim in the last outer iteration. It displays several statistics representing the population average of the score (roughly the same as the utility of a combined route/time choice model) over MATSim iterations. The yellow curve deserves most attention; it indicates the population average of the actually experienced score within a given MATSim iteration. Clearly, the curves are still slowly changing at iteration 400, suggesting that running MATSim even longer would deliver a more precise assignment.

The overall shape of these curves indicates a possibility for improving the Regent/MATSim simulation speed: One sees that the average score trajectory flatten out relatively quickly and then change only slowly. It hence may be computationally advantageous to use only relatively few MATSim iterations during the first Regent/MATSim interactions and to increase the number of MATSim iterations as the overall system converges. Further studies would be needed to make such an approach concrete.

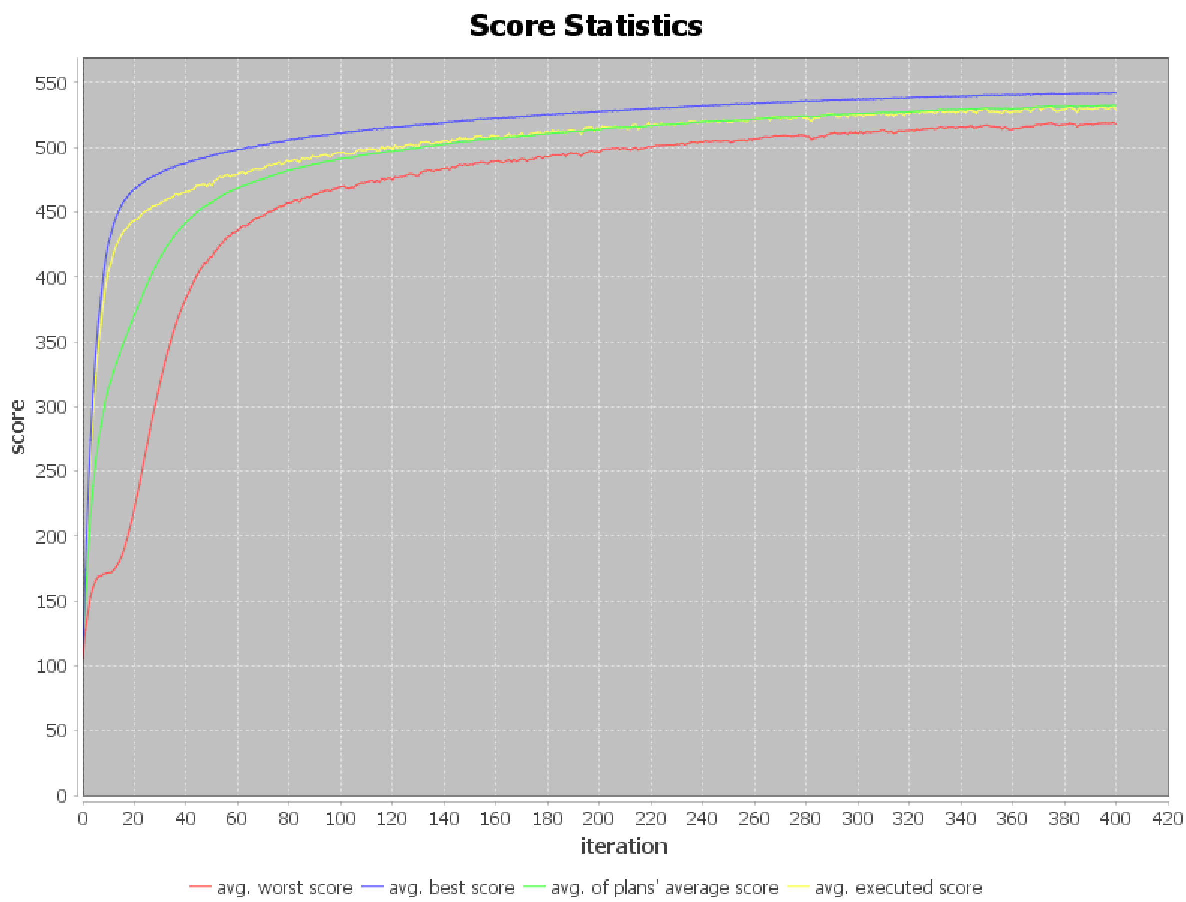

Figure 15: MATSim convergence measures over iterations

As explained before, departure time is simulated within MATSim; only the sequencing of activities is a priori defined. Figure 16 shows the resulting departure time structure per trip purpose. The x-axis represents hour-of-day, the y-axis represents number of departures per hour.

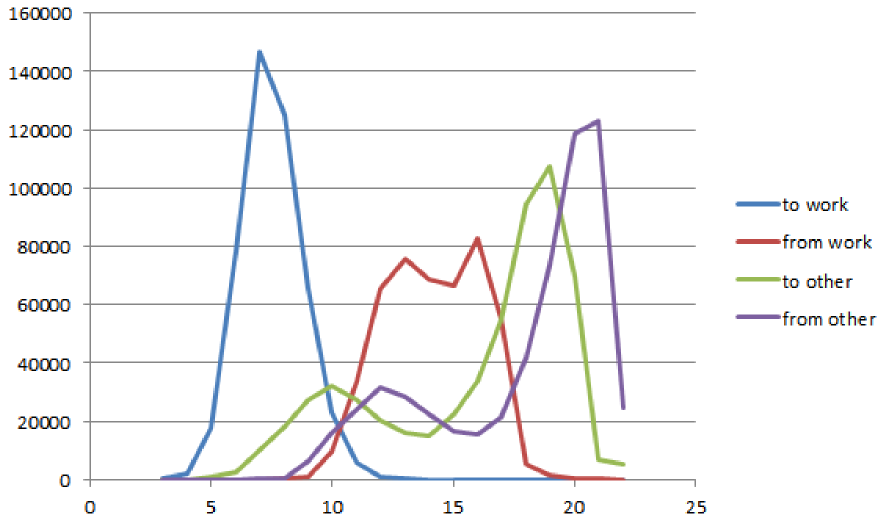

Figure 16: Simulated departure time distribution

One observes between 5am and 10am a peak of trips departing for a work activity. The travel returning from work is more spread out and has two peaks, one before and one after 3pm. This can be explained by some working individuals having to implement only a single home-work-home tour, whereas others have to implement an additional “other” tour in a “home-work-home-other-home” schedule. The latter individuals have to leave work early in order to be able to implement the subsequent “other” tour before 21:30, the closing time of “other” activities, cf. Table 6. Another interesting observation refers to the departure time distribution for traveling to an activity with purpose “other”. One flat peak around 10am and one steep peak around 7pm can be observed. This is again a consequence of different activity schedules in the population: Those having to implement only a single home-other-home tour have the scheduling freedom to select a departure time between the relatively congested morning and evening rush hour, whereas those who can do “other” only after “work” are concentrated in the later parts of the day.

To investigate the sensitivity of the system with respect to a policy measure, the Stockholm congestion toll was added to the model system, cf. Table 8. The toll was implemented in MATSim on all links that also were indicated a tolled in the TransModeler network. The following comparison is based on a single simulation without toll and a single simulation with toll. Clearly, a statistically more thorough analysis of several simulation replications would be needed for the assessment of a real policy measure.

Start time	**End time**	**Amount [SEK]**
06:30	07:00	10.0
07:00	07:30	15.0

07:30	08:30	20.0
08:30	09:00	15.0
09:00	15:30	10.0
15:30	16:00	15:0
16:00	17:30	20.0
17:30	18:00	15.0
18:00	18:30	10.0

Table 8: Stockholm toll time structure and levels

The overall effect of the toll is demonstrated in Figure 17. It shows the total flow over all tolled links as a function of time-of-day, for both the experiment without toll and the experiment with toll. The morning and evening peak are clearly reduced by the toll, meaning that the simulation system reacts in the right direction. Between the peaks, however, one observes only a very small reaction to the toll. Three possible explanations of this effect are subsequently given, both being related to the lacking calibration of the model system. (i) The relatively tight opening times (cf. Table 6) may impose too much time pressure on the simulated travelers, giving them too little opportunity to schedule their activities outside of the tolled time interval. (ii) The utility function parameters representing the trade-off between travel and money are only rough estimates. (iii) Only private traffic is simulated; commercial traffic that arises outside of the rush hour is not included and hence can also not react to the toll.

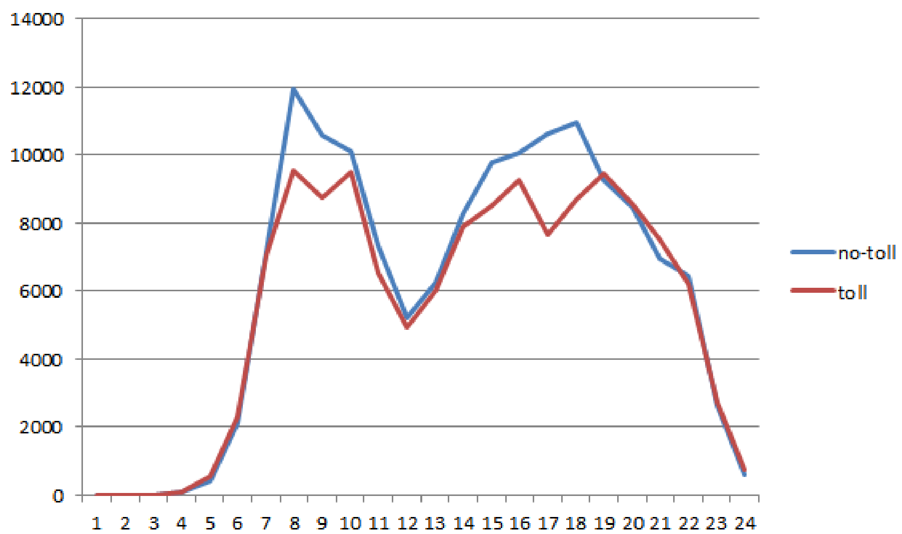

Figure 17: Total flow through toll zone

To further demonstrate that the model system is capable of capturing effects at the individual level, the population of peak-hour toll payers is analyzed. In both the no-toll and the with-toll scenario, all travelers that crossed a toll link during the time of maximum toll (as defined in Table 8) were identified. It is conceptually straightforward to go through all attributes attached to these travelers and to identify the relevant differences between them in the two scenarios. However, given the simplicity of the behavioral models used in the present, uncalibrated prototype, most attributes have no effect on travel behavior and do hence not lend themselves to a comparison.

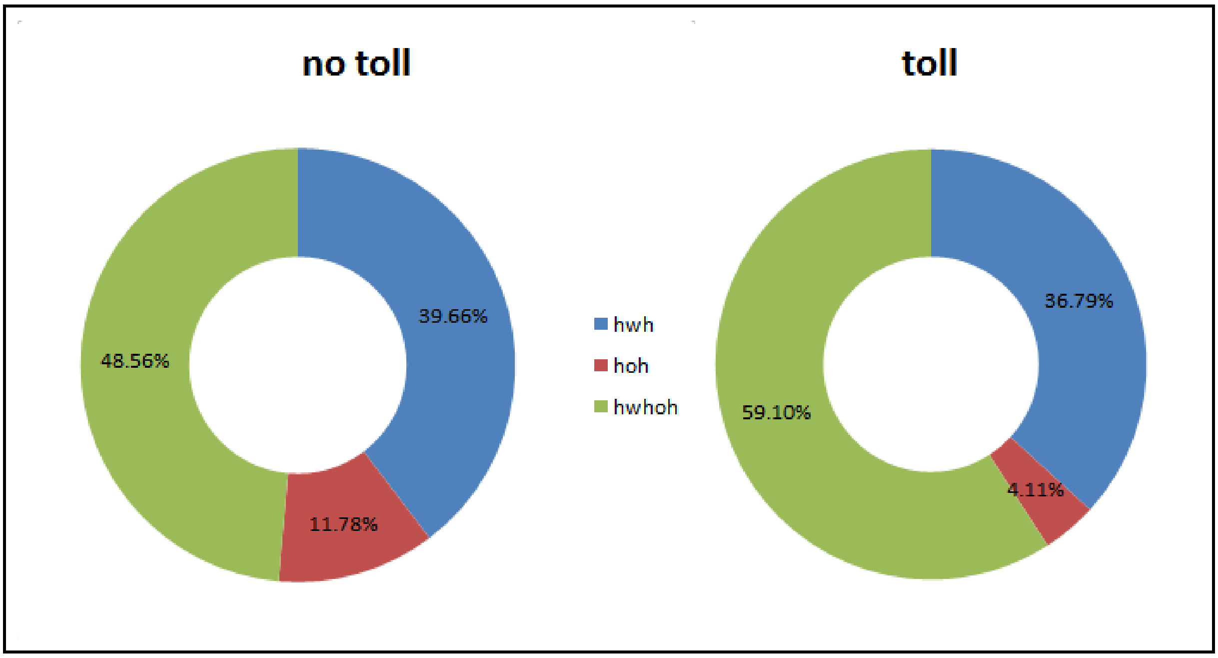

Figure 18: Activity distribution of individuals paying maximal toll

Important and plausible differences arise, however, in the distribution of all-day travel patterns of the peak-hour travelers, which are illustrated in Figure 18. (When interpreting these results, it should be remembered that relative shares of travel patterns are shown that do not reflect the total reduction of peak-hour travel in reaction to the toll, which is indicated in Figure 17.) The introduction of the toll hardly affects the share of travelers pursuing a home-work-home schedule, leads to a clear reduction of the share of travelers pursuing a home-other-home schedule, and also leads to a clear increase in the share of travelers pursuing a home-work-home-other-home schedule. The reason for this are the different opening times and planned activity durations (cf. Table 6) that make it rather easy to shift a single "other" activity out of the high-toll time window but that leave very little flexibility in adjusting a full home-work-home-other-home schedule in the same manner.
In summary, the IHOP system displays even without calibration a plausible reaction to a time-dependent toll both at the level of aggregate flows and of individual travelers.

5.3 Simulation study involving TransModeler and MATSim

It was not possible to simulate a full-day dynamic travel demand in TransModeler due to memory limitations on the deployed computer. However, it was possible to execute the program for several TransModeler/MATSim iterations before it crashed due to lacking memory. This means that it is technically possible to insert TransModeler into the IHOP system. The memory issue is currently being addressed by installing TransModeler on a workstation with a memory of 128GB. This is, however, an ongoing endeavor, and results are not yet available.

5.4 Additional simulation study involving MATSim and Scaper

Scaper is a travel demand model developed at KTH. It is based on a dynamic discrete choice model, meaning that it essentially relies on the observation that all-day travel behavior arises along a time line and hence describes all-day travel decision making as a sequential process. Dynamic discrete choice models implicitly perform a complete enumeration of all travel alternatives. As indicated in Figure 1, Scaper was also integrated in the IHOP system and is hence available as an alternative to Regent. This effort was undertaken outside of the scope of the IHOP2 project but is mentioned here to underline the flexibility of the IHOP system.

REFERENCES

Almroth A., S. Berglund, O. Canella, L. Engelson, G. Flötteröd, D. Jonsson, I. Kristoffersson and J. West (2014). Further development of SAMPERS and modeling of urban congestion CTS Working Paper 2014:10.

Caliper (2016) Transmodeler Traffic Simulation Software, Version 4.0 User's Guide.

Flötteröd, G. (forthcoming). Queueing representation of kinematic waves. In A. Horni, K. Nagel, K.W. Axhausen (editors), The Multi-Agent Transport Simulation MATSim.

Larek, P. (2015). Multi-Agent Traffic Assignment of a Synthetic Stockholm Population. Master thesis, KTH Royal Institute of Technology.

Nagel K., B. Kickhöfer, A. Horni, D. Charypar (forthcoming). A closer look at scoring. In A. Horni, K. Nagel, K.W. Axhausen (editors), The Multi-Agent Transport Simulation MATSim.

Trafikverket (2014). Trafikverkets plan för utveckling av samhällsekonomiska metoder och verktyg, effektsamband och effektmodeller inom transportområdet. Utgivningsdatum: 2014-0-14.

Zilske, M (forthcoming). How to write your own extensions and possibly contribute them to MATSim. In A. Horni, K. Nagel, K.W. Axhausen (editors), The Multi-Agent Transport Simulation MATSim.