MASTER

Assessing and visualizing the mobility effects of the Eindhoven vision 2040 for the central area of the city

an activity based multistate supernetwork approach

Maas, T.L.

Award date:
2015

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
ASSESSING AND VISUALIZING THE MOBILITY EFFECTS OF THE EINDHOVEN VISION 2040 FOR THE CENTRAL AREA OF THE CITY

An activity based multistate supernetwork approach

Maas, T. L.
S094406
SUMMARY

Recently, the municipality of Eindhoven presented the new spatial policy ‘Op weg naar 2040’. The main aim of this project is to assess the likely effects of this policy with regard to traffic flows in the central area of the city. The municipal spatial vision consists of plans to reduce the amount of cars in the inner city area and to introduce so-called High Quality Public Transport (‘HOV’) in order to create a more pedestrian friendly central city area. Therefore several roads will become car-free, but also fewer parking lots, reducing the maximum speed to thirty kilometer per hour, and fewer connections towards the ring road around the inner city, are related measures that can be extracted from the vision. One guideline is the ‘Slow Motion – Fast Forward’ principle, which stands for slow traffic where possible, and fast traffic where needed. Also the ring road will be adapted to this principle by means of two main lanes for fast through traffic and service roads on either side for local traffic, going to have an access ramp and exit function for the ring road. Existing service roads will be used more often, allowing delaying intersections to disappear. To examine whether the municipal proposals have the desired effect, an activity based transportation demand model will be used.

According to the literature study transportation models (like rule-based, utility-based or optimization) can be dynamic, meaning that the model can learn from results of previous situations, or be static, leading to equal results while using the same input. Furthermore there could be a focus on predictions for each individual (micro-simulation). The Eindhoven University of Technology developed a Multistate Supernetwork (‘MSN’), which is a static micro-simulation and optimization model. For each individual an activity diary will be created, which will be projected on the supernetwork of roads and land use locations. The least cost route is chosen based on weights given to links, like road types (like speed and fuel consumption), and general parameters (representing costs and transport mode preferences). The term ‘supernetwork’ refers, among others, to the way this model uses the ability to combine different transport modes in order to simulate realistic mobility patterns. Therefore the supernetwork can keep locations of vehicles in memory, by creating a new shadow network devoted to the node the vehicle is left, to ensure the way back can be found, without lost items. The MSN-model is ready to be tested on other projects than it was originally designed for. However, the MSN-model did not include a visualization component, but merely a textual output-file about transport mode division and intensities. As it is hard to compare the use of traffic infrastructure and transportation modes under different scenarios, an application that is able to analyze the results is developed as part of this project, making use of C++ coding, VBA, GISDK and TransCAD. The objective of the final project is therefore twofold. Testing the MSN-model in a new situation and the development of a visual component, to ease the interpretation of the output of the supernetwork model, on one hand, and assessment of the effects of the municipal spatial policy on the other hand. To be able to measure the consequences, research questions are formulated regarding the traffic intensity inside the inner city area, solvation of existing bottlenecks and preservation of new arising shortcuts, and capacity of the ring road.

The MSN-model was tested for the Rotterdam region, therefore new input files, concerning road network, Public Transport network, land use locations, individual profiles, household profiles, and a parameter file should be created. Therefore several scripts (SPSS, Python) are written to transfer information from several source data in the regulative order. Especially the Python
script for the activity program files is complex and selects information from MON/OVIN data according to predefined rules. Other data sources are NWB (Road file), REISinformatiegroep (Public Transport) and BAG (Land use). In order to attach activities to land use locations, the category division of MON/OVIN is transformed into BAG categories. For calibration of the parameters the first run of the base scenario is compared with the transport mode division and traffic intensities provided by the municipality and BASEC. Subsequently, parameter weights were adjusted, as well as some parameters inside the MSN-model, like maximum distance to public transport and corresponding search distance. Also an option is added to overcome memory issues, by saving progress every 4,000 individuals. With the suggested changes in parameter weights the transport mode division comes very close to the numbers extracted from the most recent municipal residents survey: 8% by PT, 49% by bike and 42% by car.

First the graphical component for the model is put together and tested. The output of the Multistate Supernetwork is restructured in consecutive connected time lines and location lines, preceded by a transport mode indication. The designed component is able to analyze this data, select time and locations for the prompted time frame, and splitting up into transport mode. This will lead to transport mode files, which will be loaded into a pre-created TransCAD underlayer, with predefined legend and scale. The focus in the project is on the location of cars in the network, because they influence the traffic intensity and are responsible for the traffic congestion within the city center. The model makes use of all kinds of transport modes (foot, bike, car and public transport). Regardless how many people travel by Public Transport, it will not affect the intensity of busses: busses always make their routes, independently of the number of passengers. However, to have a complete overview, besides cars and bikes, also public transport will be made visual in the visualization phase of the project.

After the calibration process, the scenario based on the municipal vision can be executed. By means of the graphical output for different time frames, the scenario can be compared with the scenario based on the current situation. One of the bottlenecks of the current situation is caused by traffic coming from the north, and leaving in the west by the Karel de Grotelaan, while traveling through the inner city area, instead of the ring road, leading to 38% more traffic in this area. Because of that the European Air Quality Norms are exceeded at the level of the Mauritsstraat. The proposals represented in the municipal scenario will lead to a calmer inner city, where new entry roads are being used as planned. However, some other roads inside the city center will be doubled in traffic, but will probably not cause new traffic congestions, except during the evening rush, where two streets need extra attention. At first sight the suggestions of the municipal vision will not lead to a change in transport mode division. However, proposals for a new more frequent HOV in combination with new P+R locations were not taken into account in this study. For bikers only changes in road types were applied, which seems not to be enough to ‘convince’ them to choose for historical routes. However, most of the cars that crossed the city center before, will now end up on the ring road, following the municipal vision. The question is whether or not the ring road can handle the increased amount of traffic. Some parts of the ring road will be so crowded that 30% of all cars driving by that road section will encounter traffic congestion. Nevertheless, because of fewer crossings the amount of traffic can be handled in a smoother motion than in the current situation. Attention should also be paid to the way service roads are connected to the ring road in order to make the traffic flow as efficient as possible, as the interaction between those lanes can influence the capacity in a negative way.
ACKNOWLEDGEMENTS

A journey, I was told, but it seems to be a road trip of almost a year for me. One of which I planned the route, but roadblocks and diversions are inevitable. I have seen wonderful things and learned a lot in this field of research. Some locations urged me to stay longer, especially when the engine broke of my transport mode. How to fix the input connections with missing documentation pages and poor internet connection? Luckily enough I met inspiring people, which I want to thank: Erik van Hal (Eindhoven municipality) for his time to discuss now playing topics and sharing traffic information, and Fred Tilburgs (Hermes) for their cooperation to provide anonymized travel details. Of course many thanks to my supervisors Aloys Borgers, Theo Arentze, Bauke de Vries, and Feixiong Liao, to help improve my journey and their widespread availability and quick response. Besides, I want to thank Luuk Visser to introduce me into the world of Python and Sacha Vervuurt for her confidence and patience. Last but certainly not least, many thanks to Peter van der Waerden for his knowledge, preparedness of his data collection and his course which introduced me into the principles of GIS and TransCAD. The end of the trip has been reached and therefore I express the hope that I can continue to devote myself to this beautiful intersection of urban planning and traffic mobility.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Objective</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Relevance</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Structure</td>
<td>2</td>
</tr>
<tr>
<td>2. Literature</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Activity based models</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Conclusion</td>
<td>13</td>
</tr>
<tr>
<td>3. Vision</td>
<td>14</td>
</tr>
<tr>
<td>3.1 Municipal Vision 2040</td>
<td>14</td>
</tr>
<tr>
<td>3.2 Conclusion</td>
<td>18</td>
</tr>
<tr>
<td>4. Approach &amp; preparation</td>
<td>19</td>
</tr>
<tr>
<td>4.1 MSN-model</td>
<td>19</td>
</tr>
<tr>
<td>4.2 Data preparation</td>
<td>20</td>
</tr>
<tr>
<td>4.3 Scenarios</td>
<td>24</td>
</tr>
<tr>
<td>4.4 Conclusion</td>
<td>26</td>
</tr>
<tr>
<td>5. Software &amp; visualization</td>
<td>29</td>
</tr>
<tr>
<td>5.1 Adjustments</td>
<td>29</td>
</tr>
<tr>
<td>5.2 Visualization application</td>
<td>30</td>
</tr>
<tr>
<td>5.3 Conclusion</td>
<td>32</td>
</tr>
<tr>
<td>6. Calibration &amp; comparison</td>
<td>33</td>
</tr>
<tr>
<td>6.1 Calibration</td>
<td>33</td>
</tr>
<tr>
<td>6.2 Scenario comparison</td>
<td>34</td>
</tr>
<tr>
<td>6.3 Conclusion</td>
<td>38</td>
</tr>
<tr>
<td>7. Conclusions, recommendations &amp; discussion</td>
<td>41</td>
</tr>
<tr>
<td>7.1 Conclusions</td>
<td>41</td>
</tr>
<tr>
<td>7.2 Recommendations</td>
<td>43</td>
</tr>
<tr>
<td>7.3 Discussion</td>
<td>44</td>
</tr>
<tr>
<td>References</td>
<td>45</td>
</tr>
<tr>
<td>Appendix</td>
<td>47</td>
</tr>
<tr>
<td>1. Inserting activity episodes into project agendas (Miller &amp; Roorda, 2003)</td>
<td>47</td>
</tr>
<tr>
<td>2. Inserting an activity into a person schedule (Miller &amp; Roorda, 2003)</td>
<td>47</td>
</tr>
<tr>
<td>4. Districts and radials (Helms, 2013)</td>
<td>48</td>
</tr>
<tr>
<td>5. Results of the Slow Motion, Fast Forward principle (Helms, 2013)</td>
<td>49</td>
</tr>
<tr>
<td>6. High Quality Public Transport (Helms, 2013)</td>
<td>49</td>
</tr>
<tr>
<td>7. Python script to reduce amount of land-use locations</td>
<td>50</td>
</tr>
<tr>
<td>8. Python script to create all activity program files</td>
<td>50</td>
</tr>
<tr>
<td>9. C++ script to analyze the MSN-output and prepare TransCAD readable files</td>
<td>56</td>
</tr>
<tr>
<td>10. GISDK script to visualize the analyzed output using TransCAD</td>
<td>64</td>
</tr>
<tr>
<td>11. Excel Visual Basic script to execute the GISDK script</td>
<td>65</td>
</tr>
<tr>
<td>12. Matlab script to check and change activity program data (Maas &amp; Feixiong, 2015)</td>
<td>66</td>
</tr>
<tr>
<td>13. SPSS script to merge all separate PT files into one large database</td>
<td>68</td>
</tr>
<tr>
<td>14. Settings of standard adobe pdf printer</td>
<td>68</td>
</tr>
<tr>
<td>15. Declaration specified in ‘declare.h’ called by the visualization application</td>
<td>69</td>
</tr>
</tbody>
</table>
Figures

1. Sketch Eindhoven situation ................................................................. 1
2. Benefits of activity-based models in comparison with trip-based models (Liao, 2013) .... 3
3. Common structure of activity based models (Bekhor et al, 2011) ........................................ 5
4. Steps of the HAGS-module of FAMOS (Pendyala et al, 2004) ........................................ 8
5. Type of constraints for consideration in Albatross (Arentze & Timmermans, 2004a) .... 10
6. Construction of person schedules (Miller & Roorda, 2003) ........................................... 11
7. Slow motion – Fast forward (Helms, 2013) ................................................. 14
8. Historical 18 Septemberplein (Weijers, 2009) .............................................. 15
9. Proposal Vonderplein (Helms, 2013) ....................................................... 15
10. Proposal inner ring Wal/Keizersgracht (Helms, 2013) ........................................... 16
11. Proposal inner ring Vestdijk (Helms, 2013) .................................................... 16
12. Detailed road network Eindhoven region .................................................. 20
13. Road Network with Public Transport Network ........................................... 21
14. Activity land use locations together with the category locations of Eindhoven ............ 22
15. First scenario, based on the Eindhoven vision 2040 ......................................... 25
16. Concept Sketch ring road ............................................................................ 26
17. Scenario Aalsterweg .................................................................................. 26
18. Current situation Aalsterweg ......................................................................... 26
19. Scenario Heezerweg .................................................................................... 26
20. Current situation Heezerweg .......................................................................... 26
21. Print screen of the execution of the MSN-model ............................................ 29
22. Print screen of the execution of the Visualization Application ......................... 30
23. Print screen of the Microsoft Excel environment .......................................... 31
24. Print screen of the TransCAD Underlayer .................................................. 31
25. Legend of both underlayer maps .................................................................... 34
26. Visualization: base scenario, car and PT traffic [7-9h] .................................... 35
27. Visualization: municipal scenario, car and PT traffic [7-9h] .............................. 35
28. Visualization: base scenario, car and PT traffic [16-18h] .................................. 36
29. Visualization: municipal scenario, car and PT traffic [16-18h] .......................... 36
30. Intensity/Capacity relation (RWS, 1999) ....................................................... 37
31. Ring Road with visualized Intensity/Capacity .............................................. 37
32. Visualization: base scenario, bike and PT traffic ......................................... 38
33. Visualization: municipal scenario, bike and PT traffic .................................... 38

Tables

1. Overview Activity-Based Models for research ............................................. 4
2. Road types division and specification in the MSN model ................................. 21
3. Recoding MON/OVIN categories to BAG activity types ............................... 23
4. New inside parameters of the MSN-model .................................................. 29
5. Existing inside parameters of the MSN-model .............................................. 30
6. Old and new parameter weights of the MSN-model ....................................... 33
7. Traffic volume numbers for calibration on sample locations ........................... 34
8. Statistical figures transport mode division different scenarios ....................... 37
1. INTRODUCTION

Recently, the municipality of Eindhoven presented a new spatial policy (Helms, 2013). One of the aims of this policy is to reduce the amount of traffic in the central area of the city. To achieve this goal, several roads will become car-free as well as the maximum speed will be reduced. Very interesting is the impact the policy will have on other roads and public transport use. Besides, a recent developed transportation demand model is ready to be tested in other situations then it original was designed for. This model does not contain a visual component to visualize the output. As it is currently hard to compare the use of traffic infrastructure and transportation modes under different scenarios, an application that is able to analyze the results will be developed as part of this project. The objectives of this project will be presented in the next section, followed by the relevance of the project and the structure of this thesis.

1.1 OBJECTIVE

The project investigates the likely effects of the Eindhoven spatial policy for 2040 on the central area of the city of Eindhoven. According to this policy, parts of the inner city should become more or less car-free with a focus on Public Transport and pedestrians. The main aim of this project is to assess the likely effects of this policy with regard to traffic flows in the central area of the city. To assess these effects, a recently developed transportation demand model, the so called Multistate Supernetwork-model (‘MSN’), will be applied (Arentze & Timmermans, 2004b; Liao, 2013). Therefore, the spatial policy will be translated into a number of scenarios. A secondary aim of this project is to ease the interpretation of the output of the supernetwork model by visualizing the output using GIS.

The municipal spatial vision of Eindhoven for 2040 (‘op weg naar 2040’) consists of plans to reduce the use of cars in the inner city area and to create so-called ‘high quality’ public transport in order to create a more pedestrian friendly central city area. Fewer parking lots in the inner city area, a maximum speed of 30 km/h on most roads in this area, and fewer connections with the ring road are related scenarios that can be extracted from the vision (Figure 1). Undeniably, this will have an impact on the daily life of people using the central city area. To be able to measure the consequences, intentional and unintentional, research questions are formulated:

- Will the policy lead to less car traffic for the region in general, less car traffic inside the inner city area, and an increase of bike traffic and public transport inside this area?
- Will the policy solve existing bottlenecks inside the inner city area, and ensure that no new bottlenecks will arise, meaning car drivers will not search for new shortcuts?
- Can the ring road handle a larger traffic intensity with regard to the proposed system of through- and local traffic, compared to the capacity according to the handbooks?
- How to extend the MSN-model in order to visualize the model’s output?
1.2 Relevance

Two main objectives can be distinguished when it comes to relevance. First of all, this project tries to assess the consequences of the interventions proposed in the municipal vision. Quite a lot of suggestions were proposed for especially the inner city area, regarding car traffic flows. It is important to have knowledge about the effects, to ensure that no traffic congestions will arise elsewhere in the city. The municipality of Eindhoven already did some basic research. By means of advanced activity-based transportation models, more advanced assessments of the effects of the policy plans may be obtained. This may be beneficial for the people living and working in the city of Eindhoven.

The second aim of the project is to add a visual component to the supernetwork model used in this study. To make the model more user-friendly, it is needed to rearrange the output into a format that can be easily interpreted: what is the location of individuals on a prompted time and what transport modes are they traveling with? To have a clear overview of all individuals in the model under different circumstances, the output has to be projected on maps by means of a geographic information system. A piece of software will be developed to extract all needed information from the model’s output file. By doing this, it becomes possible to compare different scenarios. For each scenario, the predicted traffic flows will be projected on a map. Besides, indices like road intensities will be generated to evaluate the scenarios. The visual component is essential for the use of the supernetwork model in research projects, to be able to compare the predicted traffic flows for different scenarios quickly and easily. In addition, as the supernetwork model has recently been developed, testing the model in the field can be considered relevant from an academic point of view. Thoroughly testing the model under different conditions improves the validity and usefulness of the model.

1.3 Structure

This thesis contains six more chapters, besides the introduction chapter, divided in several sections. First of all, chapter two summarizes the literature related to modern transportation demand models, followed by chapter three that will recapitulate the municipal vision for 2040. After that, the approach of this project will be discussed in the fourth chapter, as well as the preparation process. Several steps were taken to prepare all needed data for the model and create scenarios from the Eindhoven vision. Chapter five will report the activities to get the model running for the current situation in the Eindhoven region, with respect to the needed changes in the model code and the development of the visualization application. In chapter six, the results for the different scenarios will be presented and compared with each other. Beforehand, there attention will be paid to the calibration of the parameters of the model in order to make the simulated traffic flows representative for the currently existing flows. Chapter seven consists of conclusions, as well as recommendations and a discussion section.
2. LITERATURE

In this chapter the literature used to get familiar with the topic is summarized. First basic knowledge about activity based models will be shared, then different models will pass by, based on programming structure (rule-based, utility-based, optimization or micro-simulation), type (dynamic or static) and method of data use (exogenous or endogenous).

2.1 ACTIVITY BASED MODELS

“Transportation forecasting is the process of estimating the number of vehicles or people that will use a specific transportation facility in the future. Traffic forecasting begins with the collection of data on current traffic. This traffic data is combined with other known data, such as population, employments, trip rates, travel costs, etc., to develop a traffic demand model for the current situation” (Wikipedia). Starting from 2000 a number of different activity-based models for this purpose were developed, replacing the traditional four-step models, “to improve the integrity of the models, by embedding spatial and temporal interdependencies in daily activity-travel patterns” (Liao, 2013). The traditional model was ‘trip-based’ and neglected the question where people came from, as there were no relations assumed between several trips of one individual. Because “most travel is not an end in itself but a means to bridge activities that are separated in time and space” (Arentze & Timmermans, 2004a), the focus was placed on the ‘new’ activity-based models that schedule activities. From these schedules can be decided which transport mode is chosen to fulfill the trip and whether or not activities can be combined. “Trips are part of a tour, i.e. a circuit of trips starting at home […] and ending at home with the aim to conduct one or more activities involving one or more locations […]. Clearly, the trips involved in such tours are interrelated and, therefore, cannot be modelled independently. At least, the mode chain and sequence of locations should be consistent with logical and situational constraints” (Arentze & Timmermans, 2004b). This is one of the reasons for the transition from trip- to activity-based models (Arentze & Timmermans, 2004b). Besides, the relation between trips of one individual, it becomes also possible to create relations between several individuals: “Activity-based models are designed to keep linkages between travel decisions of members of a single household, as interactions among family members often affects and in many cases largely determines people’s travel” (Kochan et al, 2008). Another difference between the traditional four-step models and the activity based models, are the increase of parameters concerning the sensitivity to transportation system related factors: “to public transportation facilities or pricing measures including parking pricing, congestion pricing and fuel pricing” (Kochan et al, 2008). This method ensures “certain trips, which are linked to activities that are not so flexible are less likely to be altered under changing traffic system conditions than others” (Kochan et al, 2008).

Several types and variations of Transportation Forecasting Systems are developed because of the potential benefits of activity based modeling (Figure 2). Where utility-based models are based on “statistical analyses of activity-travel patterns” (Liao, 2013), rule-based models, as it says, follow

1. Capture of preference heterogeneity
2. Treatment of time as a continuum and a generally superior incorporation of the temporal dimension
3. Focus on trip chaining rather than individual trips
4. Recognition of linkages among various activity-travel decisions
5. Incorporation of inter-personal interactions
6. Consideration of space-time constraints on activity-travel patterns
7. Sensitive to land-use transport policies

Figure 2. Benefits of activity-based models in comparison with trip-based models (Liao, 2013)
predefined rules. 'Micro-simulation' is an additional characteristic which implies that the model has a detailed focus on individuals. Nevertheless, this can also be the only characterization of a model (e.g. HAPP 1995 and SimAGENT 2011). "The term spatial microsimulation refers to a set of techniques that allow the characteristics of individuals living in a particular area to be approximated, based on a set of 'constraint variables' that are known about the area ... spatial microsimulation can be either dynamic or static, and can include interacting or passive units" (Wikipedia).

An extra category of models could be 'optimization-models', where the model tends to optimize the current situation to a better one. This could be done, for instance, with the use of 'agents', that can develop themselves to act better each iteration. Models that are able to adjust themselves, based on new information, are referred to as 'dynamic'. One type within the optimization models is the supernetwork, which can be seen as a network of networks. "Multi-state supernetworks represent the state-of-the-art for synchronizing networks and modeling multi-faceted choices simultaneously in terms of the high choice dimensions involved" (Liao, 2013). Each individual has as many as 'virtual network' copies as transport modes and activities one can choose of, and is able to hop from one network layer (called 'state') to another.

To make another distinction between the models, there is also examined the way the model-population is composed. Most models have used Activity Travel Diary Surveys for the estimation phase of the model, except of SimAGENT. The authors of SimAGENT claim that they use their own synthetically model system, which "contains more complete data than an activity survey data, because it recreates the entire population of the research area and does not have any missing data for each person within each household" (Goulias et al, 2011). For the application phase there is the possibility to create a synthetic population based on predictions (endogenous) or use a particular part of the population originating from the survey (exogenous). Eleven of the current activity-based models were studied to learn more about the underlying structure. An overview of the investigated models can be seen in Table 1.

In particular, each model can be divided in four segments: population, activity generation, activity scheduling and activity travel choices. First all models will be discussed in general. Thereafter, the four segments of each model are introduced.

In general
The so-called activity-based approach has become increasingly popular to predict travel demand, replacing trip-based approaches. "The activity-based approach is based on the premise that transport is generated by the conduct of activities in time and space. Activity-based models predict which activities will be conducted where, when, for how long, with whom and the transport modes involved" (Arentze & Timmermans, 2004b). Besides, these models “share a
desire to replicate the sequence of decisions that leads to observed patterns of human activities and travel” (Miller & Roorda, 2003). Activity Based Models can be used for policy support, by means of highway and transit assignments that can be based on the output matrices of the models (consisting of activities and trips, including detailed information about departure time, destination and transport mode). In theory, a common structure for all models can be distinguished, which can be found in Figure 3. While interpreting the output, one should keep in mind that the observed transportation pattern is the result of “a complex decision-making process by which individuals try to achieve particular goals in the pursuit of their activities within the spatial-temporal and institutional constraints set by the environment” (Arentze & Timmermans, 2004a) and “in the context of an ever-changing physical environment, an uncertain transportation environment and multi-day variations in planned and unplanned activities” (Arentze & Timmermans, 2004a).

Including highway and transit assignments, activity based models always have a goal where the model is designed for. The SimAGENT model (Simulator of Activities, Greenhouse Emissions, Network and Travel) is designed for estimation of fuel consumption and its affiliated pollutants emission. In order to do so the model simulates activities and travel for the South California region. The output of the model is structured in such way, it can be used as input for an analyze- and visualization component. For SimAGENT this component is TRANSIMS, that is able to “generate a series of activity path records to store in an Arcview® polyline file to show the travel path on the network” (Goulias et al, 2011). The SimAGENT model is a microsimulation model that simulates every individual within each household, but also forecast growth and future land-use, measured against economic developments.

Another microsimulation model, as well as a dynamic optimization model, is MATSim. For each agent a transportation plan is developed, based on “activities, their locations and the travel legs connecting them” (Bekhor et al, 2011). Special about this model is that the model is able to improve the plans, which get a score after each simulation step, to a higher one, “in response to conditions that arose during the simulation” (Bekhor et al, 2011).

The Tel Aviv model system makes its decisions regarding the characteristics of the main tour, based on statistical values retrieved from logit and nested logit models (Bekhor et al, 2011). For instance, the availability of cars, is predicted by the chance of having them based on household characteristics. Following this model, “the model simulates for each individual the decisions regarding the main activity for the day” (Shiftan et al, 2003), which can be "work, education, shopping, other types of activity out of home, and staying at home" (Shiftan et al, 2003). After that the destination and transport mode of the tour will be determined.

A new approach is followed in the Multi-state Super Network (MSN) model. “Hereby the transport network is copied as many times as there are possible activity-vehicle states during a tour” (Arentze & Timmermans, 2004b). Each network is, so to speak, stacked on top of each other and interconnected, representing the road network and public transport network, for
each transport mode. Every individual has its own supernetwork and those supernetworks can also be linked again. Specified nodes in the road network represent stations, where a special link exists to the public transport network, as well as links that are created during the execution of the model, to represent the location of an available or parked vehicle, but also activities which still have to be conducted. “A path through the network including such a transfer link then describes a multimodal trip, that is a trip involving one or more mode switches [...]. When using an activity link or a vehicle-park link, the individual stays at the same location, but transfers from one state [network] to another” (Arentze & Timmermans, 2004b). This method makes it possible to address different costs, or weights, to each shadow-network, for de facto the same transport links. Besides, it is easy to represent a complete tour from home to home, where home is located in the original network layer and all vehicles will be returned, when the individual ‘travels’ from the particular vehicle network layer to the previous one, using the special links that were created, as there are no other links between the two network copies, towards home. Searching for the least cost route, the path-finding algorithm considers many choices about the route, location, sequence, and transport mode. Nevertheless, “the link cost functions can be defined such that solutions are sensitive to many attributes of multimodal transport systems including attractiveness of transfer stations, availability and price of parking places, and travel, waiting and transfer times” (Arentze & Timmermans, 2004b). Besides, by using parameter weights they can be adjusted easily.

From the same authors as the MSN-model, is Albatross, which is a learning-based transportation oriented simulation system, predicting activity-travel behavior, based on “theories of choice heuristics that consumers apply when making decisions in complex environments” (Arentze & Timmermans, 2004a). Albatross is a rule-based model, containing a large set of rules, that will be applied on each decision step. These rules reflect existing conditions, for example, the position of the household, concerning decisions made by individuals. “Long-term decisions made at the household level exert a strong influence on possible household activity participation patterns as the location of the residence and workplace vis-à-vis the transportation system represent the main locations of an activity pattern and are the cornerstones of decisions” (Arentze & Timmermans, 2004a). The individuals activity program is in this way created on the basis of the household activity calendar and “the desire to meet particular activity and time-related objectives” (Arentze & Timmermans, 2004a). This results in a model that can be used for predicting "long-term impacts of policy measures or trends in society” (Arentze & Timmermans, 2004a).

Strongly related to the Albatross model is Feathers, which is a model for forecasting evolutionary activity-travel of households and their environmental repercussions. The model is able to predict day-dependent activity schedules across seven weekdays for each individual, within a household context, to investigate corresponding travel demand (Lee, 2012). The goal of the agent-based model is to analyze the impact of fuel price on the traffic demand in Flanders (Kochan et al, 2008).

The Florida model ‘FAMOS’ also schedules each individual, but provides a continuous time axis. The idea behind the model, regarding activity-based modeling of travel demand is “to recognize that people pursue their activities and trips within a constrained environment [...]. Several fixed activities at fixed locations limit the activity-travel choices of an individual” (Pendyala et al,
Equal to the SimAGENT model, it is possible to visualize the output, using an integrated GIS environment, where it is possible to map either by transport mode or trip purpose.

In contrast to the other activity based models, the ADAPTS model, has a different perspective on the concept of the activity process. "To the best of the authors' knowledge, all activity-based modeling systems assume some fixed sequence for making activity decisions as in Albatross among many others. Recent data sources such as UTRACS (Urban Travel Route and Choice Survey) have shown that planning activities are opportunistically planned, during execution of a tour which could not be handled by scheduling models, where the activities are selected first, then formed into tours" (Auld & Mohammadian, 2012). The ADAPTS model is a rule-based microsimulation model that dynamically simulates activity and travel on individual level.

Equal to the Tel Aviv model, the model of Bowman and Ben-Akiva will focus mainly on the primary activity and related travel, within a large framework of tours, based on individual preferences. “The activity pattern includes the primary activity, the type of route for this activity, including the purpose and sequence of activity stops, and the number and purpose of secondary additional tours” (Bowman & Ben-Akiva, 2001). The model selects an activity pattern from all tour alternatives, based on the expected maximum utility. “The activity pattern extends the linkage beyond that of a tour-based model to include all the tours that occur in a single day [...]. The tour destination and mode choice models are estimated as multinomial logit (MNL) models with alternative sampling [...], without relying on exogenous predictions for any of the major dimensions of the activity schedule” (Bowman & Ben-Akiva, 2001). Besides, the model is able to capture interactions between individual’s decisions during a day in an activity pattern.

The 'Household Activity Pattern Problem'-model (HAPP), developed by Recker in 1995, is a first attempt microsimulation model, based on pick-up and delivery problems, occurring in the operation and logistics research. “As applied, households ‘pick-up’ activities at various locations within a region, accessing these locations using household transportation resources and reflecting interpersonal and temporal constraints, and ‘deliver’ these activities by completing a tour and returning home” (McNally & Rindt, 2008). The HAPP model is a sort of mix between rule-based and utility-based model systems, relying on mathematical algorithms and travel and activity constraints. "An estimation procedure for the HAPP objective function, based on similarity metrics to infer the relative importance of spatial and temporal factors associated with out-of-home activities, uses a genetic algorithm and positions the application of the HAPP model within a traditional demand context [...]. Nevertheless, HAPP holds great potential to be extended both as a pure activity-based framework and also as a bridge to conventional discrete choice models of travel behavior “ (McNally & Rindt, 2008).

Population Synthesis
The population used for the execution of the model can either be created synthetically based on statistics or used together with individual’s characteristics from survey data. The SimAGENT model “synthetically recreates the entire resident population of the region, provides locations for residences, workplaces and schools for each person, estimates car ownership and type as well as main driver for each vehicle, and provides other key personal and household characteristics [...], where the characteristics are provided by the US Census” (Goulias et al,
Besides, the model "includes activity types, duration of each activities and tour modes (drive alone, with passenger, being passenger, transit or walk)" (Goulias et al, 2011).

On the other hand, the FEATHERS model, extracts demographic information and information about travel behavior from a trip-based survey. Applying the model in South Korea, however, caused the problem that using individual population data about household members, income and vehicles, were forbidden for privacy reasons, whereby a part of the data still was created synthetically (Lee et al, 2012).

The FAMOS model also creates a synthetic population for the region, including personal and household information, as well as all kind of activities each person has to conduct. For the activity information, an underlying transportation survey from 1999 is used to calibrate the model components. Also information is extracted from existing model databases for the region. The division of households over the different zones of the study area is done "according to the frequency distribution obtained for the zone" (Pendyala et al, 2004). The creation of the synthetic population is done by a separate module within FAMOS, called 'HAGS'. The steps of the module, concerning both the population creation and activity generation are shown in Figure 4.

```
Figure 4 Steps of the HAGS-module of FAMOS (Pendyala et al, 2004)
```

Activity Generation
The MSN-model extracts activities from activity diaries of surveys to assign to people. Later on the sequence is determined, and where they are conducted. Also Albatross follows this principle be it that in Albatross the activities are predicted rather than taken from diary data. The Tel-Aviv model consists of an Activity Generator Unit to attach travel behavior to each person, in a more synthetic way. "As a result, each person's daily travel is fully described and includes the types and number of daily tours, the number of intermediate stops, the destinations, the modes and the time of day" (Bekhor et al, 2011).

Instead of an activity based survey, the TASHA-model uses only trip based information, in the absence of more detailed information. The result is a bunch of possible activities that might be conducted, ready to be implemented in personal schedules, by a rule-based system. The trip diary data, gathered in 1996, is used for the activity generation in two ways: "first, the database provides the base population on which the schedule model is run. Attributes of the household and person attributes are considered to be exogenous inputs into the scheduling model. Second, TTS trip data are used for generating activity episode attributes including their frequency, start times, durations and the number of people involved" (Miller & Roorda, 2003).

The activity generator of the FAMOS-model starts with determining the boundaries of person's disposable day, namely the wake-up time and bedtime, captured in a prism representation. Thereafter the time periods of fixed activities will be fitted, like work and school liabilities. The
location of these activities for workers and students are determined “using multinomial logit models of location choice” (Pendyala et al, 2004).

Where most models don't make a clear distinction between the generation phase and the planning phase, the ADAPTS model does. “The first phase of the ADAPTS model is activity generation which refers only to the decision of whether or not to plan on adding an activity of a certain type to the schedule” (Auld & Mohammadian, 2012). These activities are extracted from the recent survey data. Besides, several survey data variables are used to determine the order of activities, in the activity scheduling phase.

Activity Scheduling
For the SimAGENT model, the different activity-components (generation, scheduling and travel choices) are more or less integrated and cannot be seen separately the way other models can: while a schedule generator produces a 24 hour activity schedule for each individual, taking into consideration household plans like responsibilities towards children, work and school, at the same time a complete description is created about the displacement of each person within each household. “In more detail, activities will be random assigned to households, whereby the first and the last activity location of household members will be corrected to make sure they have the same home location” (Goulias et al, 2011).

Instead of creating advanced large schedules for a lot of different purposes, the Tel Aviv model will only focus on the two most important trips. A transport mode is assigned, which results in “mode-specific demand matrices for different periods of day” (Bekhor et al, 2011). This method will lead to a general impression, which is considered sufficient for the purpose of the model.

The scheduling model of FEATHERS, as declared, is related to the Albatross system that is discussed later on. “The scheduling is static and based on 27 decision trees, which are trained based on collected data. Decisions are based on a number of attributes of the individual, of the household and of the geographical zone” (Kochan et al, 2008). Thereafter, choices about location, mode and duration will be made using available personal characteristics.

PCATS (Prism-Constrained Activity Travel Simulator), a module within the FAMOS-model, will create Activity-Travel Records for each individual the HAGS-module has generated. A method is conceived that is able to fill all undefined time slots with activities, the so-called ‘activity engagement’, together with relevant trip perspectives. To estimate the activity duration, a separate model module, delivers default information about the activity type. Besides, the available time in the individual’s schedule is taken into account, in relation with the travel time and any subsequent fixed appointments. Decisions concerning the amount of activities individuals are participating in, are done by means of a binary logit model, where each activity type has its own model and were formulated as “functions of personal attributes and other explanatory variables” (Pendyala et al, 2004).

“The last phase of the ADAPTS framework would be the actual activity scheduling, where the activities would be added to the planned schedule and conflicts would be resolved” (Auld & Mohammadian, 2012). The model is constantly busy to solve issues for each time step, by changing the order or length of activities, or by making the decision to cancel or add one.
Where the ADAPTS model considers a dynamic activity scheduling process, in the Albatross model as in other activity-based models the sequence of scheduling decisions is predetermined. “Schedule activities with the Albatross model involves a set of interrelated decisions including the choice of location where to conduct a particular activity, the transport mode involved, the choice of other persons with whom to conduct the activities, the actual scheduling of activities contained in the activity program, and the choice of travel linkages” (Arentze & Timmermans, 2004a). Depending on the time the activities are executed, the person is bound to the limitations and scope of each transport mode. Also other constraints may play a role, which have to be taken into account by the model (Figure 5).

<table>
<thead>
<tr>
<th>Type of constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Situational constraints (different locations at the same time)</td>
</tr>
<tr>
<td>- Institutional constraints (opening hours)</td>
</tr>
<tr>
<td>- Household constraints (bringing children to school)</td>
</tr>
<tr>
<td>- Spatial constraints (particular activities impossible at location)</td>
</tr>
<tr>
<td>- Time constraints (minimum duration and limited time-frame)</td>
</tr>
<tr>
<td>- Spatial-temporal constraints (not able to be at a particular location at the right time)</td>
</tr>
</tbody>
</table>

Figure 5 Type of constraints for consideration in Albatross (Arentze & Timmermans, 2004a)

In contrast to models that assume the traveler returns home after each activity, the system of trips captured in chains ensures more activities within the same time period. “The process in which an individual attempts to realize particular goals, given a variety of constraints that limit the number of feasible activity patterns is the conceptualized form of the actual process of scheduling activities. Having identified the model constraints, individuals have to choose between feasible activity patterns” (Arentze & Timmermans, 2004a). Albatross schedules for maximally two adults within each household, in addition, it is known whether or not these persons have obligations towards children. As the activities of children are not embodied, activities with children conducted by household heads will be included. The purpose of the authors of the model is to present a so realistic as possible way of decision-making of the model’s agents. As such, “evidence suggests that individuals tend to schedule their activities in a priority based, rather than time sequential way. The schedule position and timing attributes of higher-priority activities tend to be scheduled first and, if there is space left in the schedule, lower-priority activities are considered next” (Arentze & Timmermans, 2004a). As the model only chooses the transport mode once for each trip chain, it is considered ‘determinative’ for the model’s outcome, and therefore relying on the work activity, which takes up the greater part of the day. This is also one of the advantages of the MSN-model. Besides, the choice of a vehicle for one of the two household members, affects the possibilities for the other. The main activity can be complemented with several flexible activities, which happens one after another. “A decision whether or not to add an episode of that activity is made. If an activity is added, travel party and duration of the activity are determined before a next activity category is considered” (Arentze & Timmermans, 2004a). Albatross uses a subdivision framework of time, to allocate activities to, “where the model chooses the position with the shortest time window, to maximize freedom of choice for next activities” (Arentze & Timmermans, 2004a). At the end, the possibilities to link activities to each other will be investigated by the model, meaning to either return home in between two activities or transit directly from one to the other activity. Comparing to the MSN-model, the location of the flexible activities in Albatross are also not provided by survey diaries, but rely on “a dynamic location choice-set, dependent on the time-window for the activity,
available facilities, opening times of facilities, travel times and minimum activity duration”
(Arentze & Timmermans, 2004a). A characteristic of the model is the way the activity schedules
of both adults within the household are created, as “the model takes the schedule of the partner
– as far as developed at the end of the previous step – as input to take possible interactions of
scheduling choices between persons into account” (Arentze & Timmermans, 2004a).

Totally different is the fundamental assumption for the TASHA model, that scheduling is an
“event-driven, sequential process, in which individual episodes are provisionally scheduled as
they ‘arise’ out of the personal and household projects” (Miller & Roorda, 2003). Meaning, when
a new episode is ready to be scheduled, changes in the current schedule are most of the time
required to fit in the related activities. Besides, the model proposes a ‘bottom-up’ approach, “in
contrast to the ‘top-down’ approach used in some other scheduling models in which ‘patterns of
activities’ are chosen from a large, but finite set of observed activity patterns” (Miller & Roorda,
2003). In this new approach, the individual has more influence in fitting in activities, by
adapting durations of other activities, instead of order activities in a certain way without
changing them (Appendices 1 and 2). The authors try to grasp the everlasting process of daily
scheduling in human ‘nature’. The steps of the model’s decision-making can be found in Figure
6. Moreover, it is still possible for the model to decide not to participate in the activity, when it
comes down to a packed agenda or a required major change of a certain degree, which means
the mentioned steps could not be carried out. Besides, discussing the decision whether or not to
participate in new or existing activities,, a distinction is made between projects where other
people are involved, and personal projects, as there is an obligation to not disappoint the other
(Miller & Roorda, 2003).

1) Activity episodes are generated for insertion into each project agenda based on 1996 TTS
distributions of activity attributes.

2) These activity episodes are inserted into project agendas where they are placed into a
preliminary time sequence with other activity episodes that are connected by a common
purpose.

3) Once the project agendas have been formed, person schedules are constructed by taking
activity episodes from the project agendas and adding them to the person schedule. Activity
attributes are modified and travel is added as necessary to result in a coherent consistent
schedule.

4) A ‘clean up’ algorithm is applied to fine tune the final scheduling before or during execution of
the schedule, which may allow introduction of random events, impulsive changes, further
modification, revisions, re-sorting and planning, where unrealistically short work episodes
with duration less than or equal to a 30 minute duration will be removed

Figure 6
Construction
of person
schedules
(Miller &
Roorda, 2003)

Preliminary, ”a project is defined as a coordinated set of activities tied together by a common
goal or outcome. In addition, two key attributes of a project are the project task list and the
project agenda. The project agenda is a list of specific activity episodes being considered for
insertion into the current planning period’s schedule. Activity episodes are specific occurrences
of an activity and only persist within the model until they have been scheduled and executed.
The project task list consists of activities which the person might engage in at some point in the
project, but may not necessarily do so within the current planning period” (Miller & Roorda,
2003). Like in the Albatross model, schedules of all members of the household are created in
consultation with each other, as it appears in the real world.
Activity Travel Choices
The reachability of all 15 different types of industry is captured in a data frame. Therewith the SimAGENT model has a different approach, not looking to road characteristics, but instead let it depend on type of industry, travel period, visiting hours and car ownership. The intended purpose of the model is after all the estimation of air pollution by all traffic (Goulias et al, 2011).

The goal of the MSN-model is to create a model that is able to make all available choices at once. By means of the structure of the ‘layered’ supernetwork it becomes possible to integrate “choice of activity sequence, choice of activity locations, choice of parking locations, choice of intermediate trips, choice of mode and mode transfers and choice of route” (Arentze & Timmermans, 2004). The available paths through the connected ‘virtual’ networks, by special links, represent this possibility. The least-cost-path indicates the most attractive trip chain, taken all known characteristics into consideration at once. Nevertheless, the link costs are dependent of the chosen transport mode, which all represent a copied layer of the original network. Particularly, the possibility to change transport mode during one trip, is a singularity among the activity based models. "In this way the generalized cost function and also the start and end points of a path can be defined state dependently, to ensure an optimal sequence of travelling, transferring, parking, conducting activities and dropping off products" (Arentze & Timmermans, 2004). Being aware of this, there can be stated that the activity scheduling component and the activity travel choices component are inseparably interconnected to one part, as also the location decision of an activity is based on this system of least-cost-paths.

For the FAMOS model, the duration someone is visiting a location, depends on the travel mode so he is able to make his next activity in time. Likewise, the possibility to use Public Transport facilities depends on the operating hours, affecting the availability in the choice set. The travel time is derived from a zone-to-zone database, instead of calculated based on road characteristics, like the SimAGENT model. "If the automobile or public transit is used, a zone-to-zone travel time is obtained [...] depending on the time of day (peak or off-peak). Travel times by bicycle or walking are computed using an assumed mean speed of travel and the zone-to-zone distance" (Pendyala et al, 2004).

Dynamic Aspects
‘Dynamic’ is a term with a lot of meanings. According to Oxford Advanced Learner’s Dictionary (2010) it stands for the way in which “things behave and react to each other in a particular situation”. However, in the context of this literature study it means whether the model can change itself over time through learning. So, whether the model is able to change behaviour on the basis of a previous choice (like the link-cost function of the MSN-network, which is based on the transport mode), is not captured in the term here, as almost all (rule-based) models do that in a certain way. In fact, the model has to be able to evaluate decisions to adapt the best solutions in new decisions. The dynamic aspects of each studied dynamic model, is listed below.

"A MATSim simulation converges to a state analogous to the user equilibrium through a process of systematic relaxation, by adapting and deriving a set of feasible plans for each agent from their original initial plan" (Bekhor et al, 2011). Every iteration of the model the agents try to improve their original plans. “Feasible new plans can be derived from existing ones by changing activity timings, locations, re-routing travel legs between activities, changing transport modes
connecting activities or dropping activities from the activity schedule altogether” (Bekhor et al, 2011).

“In each step of Albatross, dynamic constraints determine which choice alternatives are feasible given the current state of the schedule. History dependence of decisions is taken into account by including outcomes of previous decisions as input to each current decision” (Arentze & Timmermans, 2004a). Besides, the Albatross model claims to be flexible regarding unexpected activities. “Unlike other models, which relied on utility-maximizing theory, we assume that choice behavior is based on rules that are formed and continuously adapted through learning, while the individual is interacting with the environment (reinforcement learning) or communicating with others (social learning)” (Arentze & Timmermans, 2004a). Besides, “the learning theory on which Albatross is based implies that rules governing choice behavior are heuristic, context-dependent and adaptive in nature” (Arentze & Timmermans, 2004a).

Also the ADAPTS model can be stated as a dynamic model, as for instance “agents store the results of their actions in a long-term memory and these results are used to make future decisions” (Auld & Mohammadian, 2012). The authors of the model clarify that the dynamic aspect of the model is a “fundamental concept […] which can be influence [sic] by current needs, outside constraints and the past experiences of the individual” (Auld & Mohammadian, 2012). To be able to create a model based on the real world, the authors state that models should be dynamic and allow flexible activity scheduling, focused on the individual. Major differences can occur between models where individuals know what to do which time, in comparison with models where activities can pop-up, leading to a situation the model has to handle somehow.

2.2 Conclusion

For this project, several models discussed in the literature section can be applied, and several models are not advisable to use. The TASHA model advertises with a ‘bottom-up’ approach, which seems to be less useful, in a situation you already know which activities were performed in reality. The Albatross model is suitable for the project description, however, as many other models is not able to change transport modes during one trip. Also merely a maximum of two household heads can be simulated. The MSN-model is very promising in the way it approaches problems. A lot of characteristics can be analyzed at once by least-cost-paths through a composition of networks. However, the weakness of the model is the novelty, where not all components are fully developed or exploited, such as household interconnections. Also the ADAPTS model or the MATSIM model can be applied, with the advantages of being truly dynamic. However, with a comparison of scenarios mainly based on different road networks, rather than a larger development, where the model can actually learn and make economic decisions for the future, it is not needed to use such an advanced model. On the other hand, the FAMOS and the SimAGENT models are not accurate enough for a detailed look into a city center, as for instance road characteristics are not taken into account and travel time is calculated using zone-to-zone databases. Remain the HAPP model and the Tel Aviv model, and despite that both models probably can be developed and deployed, the in house developed MSN model is more challenging.

The supernetwork model will be applied, as the technique to simulate the city on a high detailed level in a realistic way. No doubt the MSN-model also has its downsides, as the model always
chooses the least-cost path, not taking into account other road users and household members. However, none of the investigated models is able to model travel behavior in that high level of detail and, since it is still possible to apply for changes, regarding the output structure of the MSN-model, it seems the best choice to access the traffic effects of the municipal vision (Helms, 2013). Besides, the model is able to combine different transport modes to simulate realistic mobility patterns and has already been applied in the Rotterdam region (Liao, 2013). Therefore, applying the MSN-model is also an opportunity to test the model in a new environment, contributing to improve and develop the model.

MSN is developed in 2002 and can be considered a ‘microsimulation optimization’ model, being ‘static’ and ‘exogenous’, which means the survey population used for the development of the model is also used for the execution of it. The model is not able to adjust itself based on new information, however, it tries to find the one optimal solution.
3. Vision

In this chapter the municipal vision of Eindhoven for 2040 will be discussed, regarding proposals of the traffic network, in particular the road network. Several concrete plans and principles are reviewed in next section, which were documented in ‘Op weg naar 2040’.

3.1 Municipal Vision 2040

The vision of the municipality for 2040 (Helms, 2013) is a vision for the entire city and speaks of accessibility and sustainable mobility: more walking, more biking and a better use of the Public Transport. Space has to be created for those facilities in order to create an attractive public space, at the expense of space for the car. Nevertheless, the accessibility of the city (center) must be beyond question. The principle here is ‘slow motion – fast forward’: priority to slow traffic where it is possible, priority for cars where it is needed (Figure 7).

Also an improvement of the quality of the urban environment is scheduled with more space for green areas. Not necessary paving is replaced by structural green. Besides strengthening the qualities of the inner city, the connections between ‘places to be’ get attention. After all, the traffic interventions which have the most impact on citizens and commuters, would be those for cars. The inner ring will disappear as a contiguous auto route and the maximum speed within the ring road is reduced to 30 km/h (except on new radials). Furthermore there will be fewer and easier connections to the ring road, to achieve a faster traffic flow. Instead of cars, there is plenty of room planned for Public Transport and bikes through the inner ring. Also the ambition is to connect Public Transport networks seamlessly to each other, due to a higher frequency.

To achieve a pleasant environment for pedestrians, as well as to create a faster Public Transport, a lot of overpasses will be created to reduce huge barriers. The inner city grows beyond the inner ring, as ‘De Bergen’, ‘Het Tramstraatkwartier’, ‘De TU/e-campus’ and ‘Strijp-S’ are also part of the enlarged inner city already, whereby, both in use and in perception, a distinction is perceived between both sides. The historic radials will be used for fast cycle lanes, where the car is second-class. All proposals within the municipal vision are investigated on solving capacity and effectiveness towards mobility, accessibility, economics and healthiness, according to the vision. Assumed is the existence of the vast majority of dwellings, industrial areas, facilities and infrastructure for 2040, as the future is “hard to predict”. Therefore the vision is reviewed every five years, however, it will form the basis of future projects. Moreover, the neighborhoods and the city center should be free of nuisance and inconvenience of traffic: distributor roads are underutilized. Therefore a change in sequence of transport mode is mentioned for 2025 as well to reduce car-use especially for short distance trips: increase of walking with 10%, increase of bike-use with 10%, increase of PT with 50% and decrease of car-use with 9% (not corrected for demographical growth). Concluding, Eindhoven should be a city...
where historic radials - carrying the past of the city districts - and the ring road - including service roads and modern radials - are the ideal traffic structure. Where the inner ring functions as boulevard with priority to pedestrians and bikes, good Public Transport and an optimal accessibility of parking facilities. This 'slow motion – fast forward' principle is already applied on federal roads A2 and counterpart N2. The principle can also be used on the most important road of the city: the ring road, having a connect-and-divide function. This should lead to fewer connections with the ring road, increasing the use of the already existing service roads, as the traffic flow have to be guaranteed.

The focus of the vision is at the inner city, radials and the ring road. In order to reduce car traffic inside the Ring, it is important that the Ring can handle the amount of car traffic. This project will assess the effects of the vision; whether or not actors choose for other transport modes, use shortcuts, or get stuck on traffic somewhere. Some concrete proposals for the city are discussed in the municipal vision as well. Space is created on places where the chance for success is the biggest and meaningful for the entire city, as the transformation of the 18 Septemberplein was in the past (Figure 8). The first proposal is an extension of the bus station ‘Neckerspoel’ towards the North, to accommodate the growth and to create safer and more straight connections for pedestrians between the inner city, station and TU/e Science Park.

A proposal with more consequences for car traffic is to close the road in front of the PSV stadium, to create a square with parking facilities below. This should overcome the traffic jam issues during rush hour within the inner city. Also the historical route Strijpsestraat-Gagelstraat can be recovered as part of this plan (which will form a fast connection for bikes between the inner city (Witte Dame) and Strijp-S), while the Vonderweg and the Maurisstraat can become living streets again (Figure 9). To enlarge the inner city with surrounding shopping streets (‘De Bergen’, ‘Tramstraatkwartier’), the inner ring will be closed for car traffic at several places as much as possible as well. Proposed is to close the street at the height of the Wal/Keizersgracht (Figure 10) and the Vestdijk between the Nieuwstraat and the Ten Hagestraat (Figure 11). Busses can still use the inner ring, nevertheless, they form the key contribution for movements within the inner city. It is even possible to reintroduce water in the
building environment of this central part of the city. The attractive experience of the inner city will be accomplished by a different use of the inner ring, according to the vision. That is a plausible theory, because the street is now dominated by cars, forming a barrier between the shopping areas. Besides more space for green becomes available.

From municipal research can be extracted that 6% - 15% of the cars within the City Center, don't have a destination within the inner ring. A lot of traffic enters Eindhoven from the North (John F. Kennedylaan), drive all the way up to the Central Station, take the road along the PSV stadium (Fellenoord-Vonderweg-Mauritsstraat) and leaves the inner city driving to the West through the 'Karel de Grotelaan'. 38% of the cars on mentioned roads follow this route, which forms a huge bottleneck, especially during rush hour. It is not surprising that the European Air Quality Norms are exceeded at just one place: the Mauritsstraat. In fact, drivers can make use of the much faster ring road (higher maximum speed and always two lanes), although the route is 1.4 km longer (5.2 km instead of 3.8 km), it has equal driving time outside the peak. Other bottlenecks expected to see in the results of the model will be for example the connection of the Aalsterweg with the ring road and the extension of the Karel de Grotelaan towards Veldhoven. The prediction of the likely effects on traffic while implementing the scenarios (under optimal traffic flow conditions) would be a better utilization of the ring road, however increasing use of short-cuts through the inner city can also be a result of the policy.

Other proposals consists of transforming the ‘Hoogstraat’ into a bike-street, with less priority for cars, as part of the 'slow motion – fast forward' principal (as is the 'Kruisstraat' already). On the South side of the city, the Aalsterweg will be deprioritized, and possibly become a one-way street; the same can be done for the Geldropseweg and the Stripsesstraat, where priority for cyclists and pedestrians have preference. The modern radials, like the Veldmaarschalk Montgomerylaan, the John F. Kennedylaan, the Karel de Grotelaan and the Dr. Dorgelolaan are realized in the 60’s and have to deal with the increasing car traffic from and towards the city center, leading to park roads with a lot of different green, and proved to be crucial in the development of the city in the past. These roads remain unchanged and form together with the ring road the entrance of the city, according to the 'fast forward' ideas. Besides these modern radials in North-South direction, in the West the Philiteelaan will be upgraded to a boulevard radial, as in the East the Fuutlaan will, both directly connected with the ring road. Besides, the
Kanaaldijk-zuid (towards city) and the Dirk Boutslaan (from city) can form an alternative radial, to help decrease car traffic at the Geldropseweg and increase the accessibility of the inner city. Not only physical changes within the city are discussed in the vision, also the awareness of the citizens plays an important role: as 12.5% of all civilians of Eindhoven moving by foot, still 15% of all trips shorter than 1000 meter is done by car. 25% of all movements within, towards and from Eindhoven is done by bike (2012). From the people of Eindhoven 40% choose a bike for their trips and 33% of the 55,000 train travelers, use a bike beforehand. 50% of the commuters traffic have distances below 7.5 km, which is done by car for another 50%. A third of all train passengers make use of bus facilities as well. For 2020 already 70,000 train travelers are expected by NS, which is an increase of 27.3%, leading to over 100,000 passengers for 2040. The Eindhoven vision assumes a large increase of walking for the execution of peoples trips. Besides, because of the development of P+R areas around the city there are attractive alternatives to continue trips by Public Transport. The vision will ensure that car traffic is concentrated at the main traffic routes, to guarantee the livability and safety of the residential areas.

3.2 Conclusion
The municipal vision (Helms, 2013) focusses mainly on the inner city area, by proposing fewer connections to the ring road, a car free inner ring and lower speed on most inner city roads. Also new priority main roads will be assigned, to unburden the inner city traffic intensity, as the main goal is to avert car traffic from the city center, that has a destination outside this region. Therefore, the ring road should get a ‘distribution’ function, dividing the traffic around the city. To be able dealing with an increased traffic intensity for this route, a principle of ‘slow motion – fast forward’ is introduced, existing of far-reaching service roads on both sides of the ring road for turning traffic and fast lanes for through traffic.
4. APPROACH & PREPARATION

The project will focus on the consequences of the changes of the traffic network according to the vision 2040. MSN is the activity based model used for this assessment, which uses activity programs for individuals. Location-based data has to be collected for the Eindhoven region, as the model has only been applied on the Rotterdam region. This new data has to be prepared in the right format to make sure it provides all information the model asks for. The MSN-model requires very accurately created input files, otherwise the execution will fail. In the first section, the way the MSN-model works is discussed. The next section will handle the data preparation of the input files in more detail. Decisions made during the data preparation process are described and explained. The third section deals with the specification of the scenarios in order to make it possible to validate the data.

4.1 MSN-MODEL

The Multistate Super Network Model (MSN) further developed by Liao (2013) is able to calculate trips for each individual within the dataset. First all input data have to be collected and prepared according to a detailed data structure file (Maas & Liao, 2015), which is specified in the next section. Based on weights the least cost route will be chosen: which transport mode, and - in case of car or bike - which combination of roads have to be picked. For this calculation also the availability of bike and car are taken into account, for instance whether or not the person owns these vehicles, or maybe he parked them before at a certain place, where he has to be returned to first, before ending his trip. Unfortunately, the model doesn't take households into account, and also it is not possible to be a car passenger. The least cost route is based on route cost and time, which can be specified inside the MSN-model. These values are more or less fixed and based on maximum permitted speed, current fuel costs and estimated average fuel consumption for each road category, discussed in section 5.1. One of the input files (per_parameter.dat) consists of parameters to be able to give weight to the parameters inside the MSN-model (money spent on fuel/tickets, travel time, waiting time, parking time, picking-up time), as well as capture base preferences of each transport mode, and transfer penalties. The magnitude of these weights depends on the research area and are determined in the calibration process, which is discussed in section 6.1.

Which activities an individual has to perform is stated in the activity program input file, where also an order to conduct these activities in, can be specified. If no order is specified, the most efficient order is chosen (in other words, the order having the least cost). In case of an unfixed activity location (the location is not mentioned in the activity program file) the MSN-model will choose the location from the land use input file based on previous mentioned weights, and also parking cost and location category, specified in this land use file. To deal with the strict structure of the model, script are created consisting of selection rules, for the activity program data (individual data, household data, and trip data). These rules are discussed in the next section under the heading of ‘Activity Program’. The MSN-model can be used to compare scenarios, formed in section 4.3. First the model will be calibrated on the current situation. After that the Eindhoven scenario, which consists of changes in the road network, can be executed. To do so a different traffic network input file has to be created first. To be able to have visual insight to compare the Eindhoven scenario with the current situation, instead of an output-text file, a visual component is developed, which is explained in section 5.2.
4.2 DATA PREPARATION

The supernetwork model input files consists of the road network (RWS, 2014; OSM, 2013), land use data (Kadaster 2014), public transport data (REISinformatiegroep, 2013) and zip code area data (Esri Nederland, 2013-2015). In addition, data about personal activities in the area is required as well, in order to create a population, as well as the activities they perform. This information is extracted from the activity diary survey data of MON (SCP, 2004-2009) - which is a mobility survey in the Netherlands by the Dutch Ministry of Infrastructure and Environment - and OVIN (CBS, 2010-2014) - which is the successor of MON since 2010, conducted by ‘CBS’ (Statistics Netherlands). To create a population large enough for the study area, all publications between 2004 and 2013 were used. The required data has to be prepared as input for the model, where most of the time scripts were created to handle the data files. To combine all Public Transport data a merge script was made, to get one large database. Also a script is created to reduce the number of land use locations. However, especially the MON/OVIN data needed to be treated with such care, that manually editing was not even possible. All used scripts for data preparation can be found in the appendix. Some parts of scripts are explained in this report for clarification. The data structure of the input files of the MSN model fits the structure as explained by Maas & Liao (2015). The creation of all input files should be done very accurate and according to the documentation, to overcome inconsistent or incomplete data.

Road Network

Figure 12 Detailed road network Eindhoven region (as part of a larger less-detailed network)

The NWB-data (RWS, 2014) is used as basic layer for the roads in the network. The area of the municipality of Eindhoven is selected and details like speed and one-way roads are added. Furthermore connections to other cities in the surroundings are created manually (bounded by
the cities of Tilburg, 's-Hertogenbosch, Venray and Roermond) by adding of motorways and regional highways. The maximum speed is provided by Open Street Map-data (OSM, 2013) by tagging the data on street name, however, different line segments of the same road often have not the same maximum speed. Therefore the roads are checked and edited manually. The network requires the specification of four types of roads, which are given own characteristics within the model. Because the original model was prepared for the regional-level, 30km/h roads did not play a role in it, while provincial and national-roads did. The Eindhoven model needs a more detailed subdivision of road types, based on the goal of this project: assessment of a municipal vision. Therefore a new subdivision is made, based on the five types of roads in the Eindhoven area. To distinguish bicycle paths a sixth category is added, where no cars are allowed (very high penalties, by reducing the speed and enlarge fuel consumption). The same thing is done for bikes: they cannot make use of the fourth and fifth category of roads. This new subdivision is implemented in the MSN model (Table 2).

Table 2  Road types division and specification in the MSN model

<table>
<thead>
<tr>
<th>Subdivision</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Speed</td>
<td>car</td>
<td>30 km/h</td>
<td>50 km/h</td>
<td>70 km/h</td>
<td>80 km/h</td>
<td>120 km/h</td>
</tr>
<tr>
<td></td>
<td>bike</td>
<td>14 km/h</td>
<td>16 km/h</td>
<td>17.5 km/h</td>
<td>0 km/h</td>
<td>0 km/h</td>
</tr>
<tr>
<td>Locations</td>
<td>Residential</td>
<td>-Radials</td>
<td>-Region roads</td>
<td>-Ring road</td>
<td>-Access roads</td>
<td>-Motor ways</td>
</tr>
<tr>
<td></td>
<td>-Inner ring</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Furthermore the one-way roads are marked, using Open Street Map data as well. However, this data does not indicate the direction of the roads, so this also has to be done manually. Given the level of analyses in this project, one-way roads have to be taken into consideration. However, the MSN model does not have to be changed for the one-way function addition. In total the road network consists of 7,405 nodes and 10,674 undirected links, including 2,475 one-way streets. The sum of the travel possibilities will lead to 18,873 directed links (Figure 12). In order to make sure all links are properly connected, TransCAD provides a node-check tool.

PT timetable

The PT input data files can be created based on open source data files (REISinformatiegroep, 2013). However these files have to be merged to get one solid database. A SPSS script was created to merge all text-files containing stop location information, stop time information, route information (global) and trip information (connections). Double cell values of the ‘stop_id’ column of ‘stops.sav’ were deleted beforehand to overcome merge problems (Appendix 13). Thereafter, all non-relevant routes/companies (except ‘Hermes’, shown as ‘Connexxion’, ‘Arriva’ and ‘NS’), all duplicate cases (caused by different timetables), as well as all trips after 24:00h were deleted.
All nearby villages are covered by the bus now, as well as the train connections, which form an important conveyance, especially between the bigger cities (Tilburg <-> Eindhoven, 's-Hertogenbosch <-> Eindhoven and Roermond <-> Eindhoven). The train is an alternative for car travel to commuters from those cities. Long distance travelling by bus is not taken into account, because the train is already an alternative: commuters can travel to the nearest station, while it does not affect the car density in the city center of Eindhoven by that decision. After the scripts also Excel had to be used to combine arrival and departure lines, leading to 2,753 stop locations and 179,245 connections (possibilities to travel between the stops), during one day (Figure 13).

Land use

Information provided by BAG data of Eindhoven (Kadaster, 2014) is used for the land use input file. The land use input file consists of location, activity type, type of parking, floor space, postcode and parking costs information. The activity types used in BAG are: office, store, industry, meeting facilities, health care, lodging, education and sport facilities (Table 3). Four location categories are distinguished for Eindhoven: 'city center', 'medium', 'airport' and 'other'. For each type specific characteristics, like parking costs, are defined in the model. Average parking costs are provided by the website of the municipality: values have been determined per parking type. The different areas are determined manually by means of TransCAD (Figure 14). The BAG data consists of Latitude/Longitude coordinates, but they are not suitable for the MSN model. Therefore a postcode data file with X,Y-coordinates will be used to tag locations to the center of the six digit postcode areas, as it is not needed that all buildings have an unique
location. The data consists of 7,179 activity locations, which is reduced to 3,564 locations by combining the same activity types within the same 6 digit postcode area, using a Python script (Appendix 7). After that activity locations in the whole Netherlands are added. For each existing four digit postcode a work location is added and attached to the nearest node of the road network, to make sure that especially this main trip is simulated well, meaning that the trip will not be relocated and therefore bounded by the Eindhoven region, because of missing locations. This leads to the total number of land use locations at 11,692.

Activity Program
To create the model population files (activity program, individuals’ profiles and household attributes), diaries from persons living in the Eindhoven region or conducting activities in this region will be selected from MON/OVIN databases. All sequences of activities taking place outside the Eindhoven region will be filtered out, except the trips of individuals that have their homes in this region. Trips with missing information, will be dropped as well and not be taken into account. A script is written using Python to get all data from the SPSS-files, and put them in the right order of the input file (Appendix 8). Some data is being recoded by the script, because the model uses other or opposite coding. In case of car availability the script assumes that an individual has a car if the household possesses at least one car AND is the main user of the car OR according to the MON/OVIN data used a car for the trip. This last statement is to overcome coordination problems, as the person could have joined a common drive, being car passenger.

Also recoding was needed to connect the MON/OVIN data with the land use data. The land-use activity type division, provided by BAG, will be used to map the MON/OVIN data into, by the Python Script (Table 3). Inside the script several MON/OVIN categories are recoded to a work location instead (which is a randomly picked office or industry location), to represent a travel activity to a particular location, as these work locations are represented in all of the four digit postcode locations, as discussed in the previous section.

<table>
<thead>
<tr>
<th>MON#</th>
<th>Explanation MON</th>
<th>Recode to BAG#</th>
<th>Explanation BAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Towards home</td>
<td>Trip dropped</td>
<td>Embedded in MSN</td>
</tr>
<tr>
<td>2</td>
<td>Work location</td>
<td>Random 1 OR 3</td>
<td>Office OR Industry</td>
</tr>
<tr>
<td>3</td>
<td>Business visit</td>
<td>4</td>
<td>Meeting facilities</td>
</tr>
<tr>
<td>4</td>
<td>Transport profession</td>
<td>-&gt; to work location</td>
<td>Not covered</td>
</tr>
<tr>
<td>5</td>
<td>Transport people</td>
<td>-&gt; to work location</td>
<td>Not covered</td>
</tr>
<tr>
<td>6*</td>
<td>Transport goods</td>
<td>-&gt; to work location</td>
<td>Not covered</td>
</tr>
<tr>
<td>7</td>
<td>Education OR Course</td>
<td>7</td>
<td>Education</td>
</tr>
<tr>
<td>8</td>
<td>Shopping</td>
<td>2</td>
<td>Store</td>
</tr>
<tr>
<td>9</td>
<td>Visit OR Stay</td>
<td>6</td>
<td>Lodging</td>
</tr>
<tr>
<td>10</td>
<td>Touring OR Promenade</td>
<td>-&gt; to work location</td>
<td>Not covered</td>
</tr>
<tr>
<td>11</td>
<td>Sport OR Hobby</td>
<td>8</td>
<td>Sport facilities</td>
</tr>
<tr>
<td>12</td>
<td>Other leisure</td>
<td>-&gt; to work location</td>
<td>Not covered</td>
</tr>
<tr>
<td>13</td>
<td>Personal services</td>
<td>5</td>
<td>Health care</td>
</tr>
<tr>
<td>14</td>
<td>Other purpose</td>
<td>-&gt; to work location</td>
<td>Not covered</td>
</tr>
<tr>
<td>14*</td>
<td>Go with supervisor</td>
<td>-&gt; to work location</td>
<td>Not covered</td>
</tr>
<tr>
<td>15*</td>
<td>Other purpose</td>
<td>-&gt; to work location</td>
<td>Not covered</td>
</tr>
</tbody>
</table>

Table 3 Recoding MON/OVIN categories to BAG activity types
*) OVIN category differs from MON category, only available in OVIN database

Whether or not the activity location is extracted from the MON/OVIN depends on the activity type: offices, industries, education and sport facilities are determined fixed. In that case a random location is chosen by the script, that matches activity location (4 digit postcode) and activity type, provided by the MON/OVIN data. When the script could not find a match, a
random location is chosen by the script only matching activity type, to prevent an error of the MSN-model (which insists that the location of a fixed activity type should be specified). If an activity is ‘not-fixed’ the activity location only matches activity type, which is done inside the MSN-model itself. To find these locations the Python script and the MSN model make use of the land-use file. The Activity Program data, extracted from MON 2004-2009 and OVIN 2010-2013, contains 12,844 individuals living in Eindhoven (3,707 individuals) or participate in activities in this area (9,137 individuals), conducting 34,848 activities all together. By using the weight factor – inside the Python script – provided by MON/OVIN (and divided by arbitrary number ‘80’) the population is scaled towards 56,368 individuals (18,038 individuals from Eindhoven), conducting 152,823 activities. The MON/OVIN weight is applied in order to create a representative composition of the population and to be able to vary the residential location of each multiplied individual, as only the four digit postcode location is provided by MON/OVIN. The Python script will randomly attach six digit postcode locations instead to each multiplied individual, extracted from the land-use file. If there is no land-use location available, the nearest node to the center of the four digit postcode location will be picked, which is already included in a summary file (‘pc4.txt’), together with the corresponding coordinates. The arbitrary number is applied in order to reduce the total number of individuals so the MSN-model can handle the quantity, because the MSN-model will take too long executing the 4.5 million individuals. Besides the amount of individuals is not representative for the amount of individuals conducting activities in this region. Later on all, during the calibration phase, the MSN output can be multiplied again to create a situation which conforms to the observed one, which is discussed in section 6.1. For the final check of the activity program files, created by the Python script, a Matlab script is used (Appendix 12), to combine the same sort of activities, lower durations over 800 minutes, reduce the amount of activities to a maximum of six and re-index the person numbers (Maas & Liao, 2015). The parameter file can be the same as in the Rotterdam model, which was based on underlying research (Arentze & Morlin, 2013), however, it can be necessary to calibrate these parameters to the Eindhoven situation during the first tests of the MSN model of this research.

After the population files are created by the Python script the composition of the population is known and can be compared with information from literature. It appears that 53% of the individuals is ‘men’ and 47% is ‘female’, with an average age of 39 years old. The highest level of education division is: 13% none or younger than 15 years, 9% Elementary School, 21% Lower Vocational School, 29% Secondary Vocational School, 26% Higher Professional Education, and 3% other or unknown. Of all people 64% possesses a car (by the given definition) and 96% individuals have a bike available. Currently, 39% is listed as a fulltime worker. It is not possible to calculate the average income, because MON and OVIN use different distributions here.

Summarized, this population exists of people living in the Eindhoven region (32%), or participating in activities carried out in this region (68%). According to the activity profiles, extracted from MON and OVIN data, can be stated that 44% of the trips are executed by car, 2% by Public Transport, 39% by bike and 14% by other transport modes. Looking to only Eindhoven citizens, 40% of the activities are conducted by car, 2% by Public Transport, 41% by bike and 17% by other transport modes. As the activities were corrected by a weight factor for individuals, rather than a weight factor for activities, it is not sure the figures can be taken as a guideline, however they can be used to frame the municipal provided values.
4.3 SCENARIOS
A few scenarios will be created in order to evaluate the municipal vision. The base scenario represents the currently existing situation, and can be calibrated using road intensities extracted from BASEC - a traffic volume research program by commercial business ‘Dufec’ (Dufec, 2015). Additional scenarios represent interventions in the traffic network discussed in the vision. The effects on the traffic flows and the use of the central city area will be assessed for each of the defined scenarios, by showing the traffic flows on a thematic map where critical points in the traffic network will be highlighted. Indices like road intensities will be used to evaluate the scenarios. The focus for each of the scenarios will be on car traffic, particular to conclude whether or not car traffic can be reduced within the city center and the region in general, however, also the effects the scenarios have on use of Public Transport and bike traffic will be visualized. The first scenario (Figure 15) is directly extracted from the vision, to come as close as possible to the suggested interventions for roads. Summarized, in this scenario the number of connections to the ring road decreases, the maximum speed at historical roads decreases, the inner ring will be car free at two different road segments, and the route along the PSV-stadium will be cut off, as well as the route in front of bus station ‘Neckerspoel’ (Appendix 3). Other scenarios can be extracted from the first scenario to investigate possible effects in more detail. Some detailed plans are added (Appendix 4, 5 and 6), in order to form scenarios.

More concretely, in the first scenario several connections with the ring road disappeared, in such a way that the service roads parallel to the ring road serve as an access ramp (Figure 16). By doing this, cars will experience an improved traffic flow, caused by fewer intersections. Some examples of the applied solutions will clarify the principle (Figure 17 and 19), in comparison with the current situation (Figure 18 and 20). However, this is only possible for one direction: inside the ring road an access ramp can be created clockwise, while outside the ring road the access ramp can only be established counter clockwise. Therefor the locations to apply this mechanism - not mentioned in the municipal vision - were selected very carefully and
deliberately, in such a way that the car driver has several options to enter the ring road: through the access ramp or by the route of a still existing crossing nearby.

In case of the Hoogstraat an underpass for cars seems to be the right solution to maintain the historical value and accessibility of both sides, with a view to shops located in this street on both sides of the ring road. On several other spots an underpass, crossover or level crossing also needs to be established for bike and pedestrian traffic, in order to guide slow traffic to specific routes (historical radials) other than for car traffic (modern radials).

The service roads that get the access ramp function, exists already most of the time in the current situation, but has less length and most of the time is only used to connect one or two side roads. The service roads in the new situation, proposed in the municipal vision, will connect more side roads and will be extended if needed. In case a crossing disappears the service roads on both sides of the street will be extended to connect side roads over a larger distance.

Figure 16 Concept Sketch ring road (own work)

Figure 17 Scenario Aalsterweg
(with access ramp principle and crossover for slow traffic in purple)

Figure 18 Current situation Aalsterweg

Figure 19 Scenario Heezerweg
(with access ramp principle and crossover for slow traffic in purple)

Figure 20 Current situation Heezerweg
4.4 Conclusion

The MSN-model makes use of several types of files, applied to the study area: the road network for each scenario, where the base is extracted from the ‘National Road File’ (NWB), the activity/land-use locations, extracted from ‘Basic Registry Addresses and Buildings’ (BAG), public transport data (REISinformatiegroep) and person/activity data, according to ‘Mobility Investigation Netherlands’ (MON) and ‘Research Movements in the Netherlands’ (OVIN). The creation of all input files should be done very accurate and according to the documentation.

The activity program data provided by MON/OVIN is recoded inside the preparation Python script, to control the accessibility of cars, and also to translate the MON/OVIN coding towards BAG coding, to be able to connect activities to land use locations. The population is scaled by a person weight factor in order to create a representative composition of the population and to vary the living locations, as the MON/OVIN data only contain a four digit postcode. This four digit postcode is randomly transformed by the script to a six digit postcode. Because the MSN-model cannot handle too much individuals, the weight factor is reduced in size, which can be corrected in the models’ output again during the calibration phase.

The scenario is extracted from the municipal vision. Summarized, in this scenario the number of connections to the ring road decreases, the maximum speed at historical roads decreases, the inner ring will be car free at two different road segments, and the route along the PSV-stadium will be cut off, as well as the route in front of bus station ‘Neckerspoel’. Because of disappearing crossings, the service roads, which most of the time already exists, will get an extra function as being access ramp (or exit road). This was already possible, but there was no need to make use of it, as there was always a crossing nearby. In the proposal there will be less alternatives. Which crossings disappears was not mentioned in the municipal vision, and therefor selected very carefully, to retain good accessibility in both directions towards and from the ring road.
5. SOFTWARE & VISUALIZATION

After the creation of all input files the MSN-model can be executed to produce the output files about the location and transport mode of each individual. However, first the model had to be extended to make this possible. Besides, the user-friendliness of the model is improved by a better interaction between the MSN-model and the user. The ‘Adjustments’ section will briefly summarize these changes. Also parameters that can have influence on the output will be mentioned and will be updated where needed. In the next section the visualization application that is able to visualize the output of the MSN-model, will be run through.

5.1 ADJUSTMENTS

The MSN model does not contain a visual component yet. For this project, it is needed to rearrange the output of the MSN model into a format that can be easily interpreted: the output has to be projected on maps by means of a geographic information system. A visualization application is written in order to present the MSN output file geographically. For this purpose, the output structure of the MSN-model is adjusted to be able to provide output for each individual according to the structure, and where X is the number of the individual.

even line individual X: “[transport mode] [location 1] [location 2] [location n]”
odd line individual X: “[transport mode] [time 1] [time 2] [time n] [time n+1]”

In this structure an individual is on location 1 between time 1 and time 2. The transport mode should be equal for both lines, and could be: 0=car, 1=bike, 2=public transport. The output of all transport modes will be combined by the MSN model to one file, in order to analyze the output by using the visualization application. Besides this important change, some small changes in user-friendliness are added. The scenario the user wants to execute is prompted, and also the different steps the model runs through are shown, as well as the progress of the model while creating the population. The size of the population can be set in order to test the model and the model will save and reload after each 4,000 individuals now to overcome memory issues (Figure 21). Some parameters were changed or added to fit the current situation into the MSN-model, regarding the new subdivisions of the road network, like speed and fuel consumption (Table 4). The table shows also the weight value number that has influence on the parameter. Some extra inside parameters, which were already set, can be found in Table 5. A few parameters were adjusted in order to be able for individuals living outside Eindhoven to conduct fixed activities inside the Eindhoven region, which they do according to MON/OVIN. This is because otherwise they fell outside the search area, leading to match errors in the model.

Table 4 New inside parameters of the MSN-model

<table>
<thead>
<tr>
<th>Public Transport</th>
<th>w</th>
<th>Intercity Train</th>
<th>Stop Train</th>
<th>Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>€/minute</td>
<td>15</td>
<td>0.30</td>
<td>0.20</td>
<td>0.10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Road type</th>
<th>w</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel use</td>
<td>14</td>
<td>1:10</td>
<td>1:15</td>
<td>1:18</td>
<td>1:20</td>
<td>1:10</td>
<td>1:2</td>
</tr>
<tr>
<td>Speed car</td>
<td>8</td>
<td>30</td>
<td>50</td>
<td>70</td>
<td>80</td>
<td>120</td>
<td>1</td>
</tr>
<tr>
<td>Speed bike</td>
<td>3</td>
<td>15</td>
<td>15</td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distance to PT stop</th>
<th>&lt;5,000 m</th>
<th>5,000-20,000 m</th>
<th>&gt;20,000 m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce</td>
<td>not</td>
<td>to 5,000 m</td>
<td>25% of total</td>
</tr>
</tbody>
</table>

Figure 21 Print screen of the execution of the MSN-model
5.2 Visualization Application

A new C++ visualization application is written that can read and analyze the data from the MSN-output file (Figure 22). The application has to be run as many times as there are scenario’s. First a file name has to be entered and a timeframe (of start time and end time) is prompted, in which research has to be conducted. In the next phase all the data is collected from the file and turned into variables (string Array). Then the prompted time period is compared with all the time values in the file (whether or not the prompted time fits between two – consecutive – time values). The model can handle more than 7,5 million time values and 7,5 million location values, divided over 300,000 lines (modifiable), which is enough for more than 50,000 individuals. After that the right location values and transport modes are searched by all matching time values. These matches are stored in a main output file and in sub output files, where each transport mode has its own sub output file. In this sub output file the amount of traffic on the same road section is summed up, so it can be easily imported into TransCAD, a GIS. Also a transport mode division of the input file and output file will be calculated. The program code for this application can be found in Appendices 9 and 15.

The next step will be the execution of a macro script by the application to import this data automatically into the TransCAD underlayer, and produce a picture of the situation. For this purpose Microsoft Excel will be launched, that can handle GISDK macro’s of TransCAD (Figure 23), since it is very complex to run TransCAD from C++. In the Excel environment the base scenario or one of the other scenarios has to be identified in order to make sure the right underlayer will be used. This layer is especially designed for the visualization purpose, and can

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>New Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk speed</td>
<td>5 km/h</td>
<td>Unchanged</td>
</tr>
<tr>
<td>Drive margin</td>
<td>30,000 m</td>
<td>50,000 m</td>
</tr>
<tr>
<td>Search distance to fixed activity location from previous</td>
<td>7,500 m</td>
<td>Unchanged</td>
</tr>
<tr>
<td>Search distance to activity location from home</td>
<td>40,000 m</td>
<td>400,000 m</td>
</tr>
<tr>
<td>Walk preference distance before bike</td>
<td>400 m</td>
<td>Unchanged</td>
</tr>
<tr>
<td>Walk preference distance before car</td>
<td>800 m</td>
<td>Unchanged</td>
</tr>
<tr>
<td>PT preference distance before car</td>
<td>30,000 m</td>
<td>50,000 m</td>
</tr>
</tbody>
</table>

Table 5 Existing inside parameters of the MSN-model

---

Figure 22 Print screen of the execution of the Visualization Application
be created by adding a column named ‘MSN’ to the ‘node’ layer and to the ‘PT’ layer. Furthermore the thematic maps are pre-created, as well as the road representation (Figure 24).

In addition, the results are zoomed exactly in accordance with the underlayer map. The script of the GISDK macro to create the pdf-pictures can be found in Appendix 10. Changes to this script have to be loaded into TransCAD before first use (using tools-add ins-GISDK-compile to UI). Be aware that for each scenario an own Underlay map is needed, if changes in the road network were applied. In this example the roads with a higher allowed maximum speed are bigger and black, while the 30 km/h roads have got a grey color. The legend of the map should be prepared in the Underlay map as well, based on the kind of time periods the visualization application is set on. The map is updated automatically with the new density numbers provided from this application: addition of a column named ‘MSN’ to the PT-layer and the Node-layer of the Underlay map is enough to make the script work. Of course the working directories mentioned in the Excel Visual Basic Macro (Appendix 11) should be updated first. Also the Adobe PDF Printer should be set as standard printer of your computer before executing the TransCAD-script (Appendix 14). The C++ code can be easily integrated with the MSN model.

The focus in the project is on the location of cars in the network, because they influence the traffic intensity and are responsible for the traffic jams within the city center. The model makes use of all kinds of transport mode (foot, bike, car and public transport). Regardless how much people travel by Public Transport, it will not affect the intensity of busses: busses always make their routes, independently of the number of passengers. However, to have a complete overview, besides cars and bikes, also public transport will be made visual in the visualization
phase of the project. Unfortunately, it is currently not possible to visualize pedestrians, because one travel as the crow flies, regardless of the traffic network (travel time between two desired points is calculated as a straight line), so do not fit the output-file structure. As a result each GISDK script will create two maps: one representing bike together with Public Transport, and one representing car together with the same Public Transport values. Because cars and bikes make use of the same road nodes, it is hard to combine both maps into one overview, as one layer will cover the other layer. However, the Public Transport intensity is included in both maps.

5.3 CONCLUSION
The MSN model has been extended with a visualization component to project results on a TransCAD map (named GrApp.map), which can be given an own lay-out, based on the different road-types. For each scenario an own underlay map is needed, if changes in the road network were applied. The Visualization Application 'Datatransfer' is divided in several sub-parts, in order to make the code as efficient and compatible as possible. The first and largest part is written in C++ code and therefore it can be easily integrated with the MSN model itself. A piece of Visual Studio code will be executed right after the C++ code, in order to create a suitable environment to communicate with TransCAD. These GISDK communication is launched by Visual Studio through Microsoft Excel. Besides the visualization extension, also some changes in the MSN model itself were applied, in order to make the output more user friendly (explanation of input files and less vague numbers on screen) and more possibilities for user input. Inside the model some parameters were changed to fit the project (road type parameters, distance to public transport if not nearby, maximum search distance), as well as the option to save progress each 4,000 individuals, to overcome memory issues.
6. CALIBRATION & COMPARISON

In this chapter the results of the MSN-model will be examined. Because the supernetwork model originally was developed for the Rotterdam region, the model’s parameters were fitted for the region of Rotterdam. Criteria based on statistics, like car-use, bike-use and PT-use, will help calibrating the parameters. The statistics of car-use and bike-use are extracted from BASEC for a small amount of representative road segments, provided by the municipality. With these traffic intensity numbers the agreement between observed and predicted traffic flows can be optimized. During the calibration process the base scenario will be compared to the real situation, in order to equalize both, by adjusting parameters. In the next section the current situation scenario will be compared with the municipal scenario for different time periods to find out which effects the municipal plans have on the inner city of Eindhoven and beyond.

6.1 CALIBRATION

According to the residents survey ’Monitor Verkeer en Vervoer’ (Eindhoven, 2014) 6% of the activities in Eindhoven is done by public transport, while 12% is done by foot, 43% is done by bike and 38% is done by car. While transport by foot is not measured in the visualization application (individuals by foot will not be visualized however it is possible for the MSN model to choose this transport mode), the percentages are corrected into percentages of visualized individuals: 7% by PT, 49% by bike, 44% by car. These numbers, as well as the total amount of individuals on several sample locations need to be reached. Therefore the traffic intensity on several roads and bus stops will be compared. According to the same investigation, when it comes to visiting the city center, 23% of the participants are open to take the bus once or more, 62% consider to take the bike instead and 34% think about taking the car as well. So under the right circumstances a lot more people are willing to take Public Transport or bike instead. The scenario comparison in section 6.2 will tell whether the proposals of the municipal vision have effect on the transport mode division. First the transport mode division had to be correct, thereafter the output can be scaled up.

At the first run the transport mode division is not good enough, as 20% of the trips are done by Public Transport, 47% by bike and 31% by car. Therefore the parameter weights of the ‘per_parameter.dat’ file have to be changed in such a way that Public Transport becomes less attractive and car more attractive (Table 6).

<table>
<thead>
<tr>
<th>W#</th>
<th>Main transport mode</th>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Waiting time activity</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>Travel time PT</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>Travel time walk</td>
<td>0.115</td>
<td>0.1</td>
</tr>
<tr>
<td>3</td>
<td>Travel time bike</td>
<td>0.08</td>
<td>0.1</td>
</tr>
<tr>
<td>4</td>
<td>Travel time bus short dist.</td>
<td>0.065</td>
<td>0.3</td>
</tr>
<tr>
<td>5</td>
<td>Travel time bus long dist.</td>
<td>0.07</td>
<td>0.35</td>
</tr>
<tr>
<td>6</td>
<td>Travel time train short dist.</td>
<td>0.055</td>
<td>0.25</td>
</tr>
<tr>
<td>7</td>
<td>Travel time train long dist.</td>
<td>0.049</td>
<td>0.25</td>
</tr>
<tr>
<td>8</td>
<td>Travel time car short dist.</td>
<td>0.044</td>
<td>0.01</td>
</tr>
<tr>
<td>9</td>
<td>Travel time car long dist.</td>
<td>0.079</td>
<td>0.02</td>
</tr>
<tr>
<td>10</td>
<td>Parking time car</td>
<td>0.075</td>
<td>0.02</td>
</tr>
<tr>
<td>11</td>
<td>Picking-up time car</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>12</td>
<td>Parking time bike</td>
<td>0.03</td>
<td>0.025</td>
</tr>
<tr>
<td>13</td>
<td>Picking-up time bike</td>
<td>0.02</td>
<td>0.015</td>
</tr>
<tr>
<td>14</td>
<td>Money spent on fuel</td>
<td>0.098</td>
<td>0.025</td>
</tr>
<tr>
<td>15</td>
<td>Money spent on tickets</td>
<td>0.205</td>
<td>0.725</td>
</tr>
<tr>
<td>16</td>
<td>Transfer per time</td>
<td>0.12</td>
<td>7.250</td>
</tr>
<tr>
<td>17</td>
<td>Access weight bike</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>18</td>
<td>Base preference bike</td>
<td>0.44</td>
<td>0.5</td>
</tr>
<tr>
<td>19</td>
<td>Egress weight bike</td>
<td>-0.055</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>Base preference P+R</td>
<td>0.48</td>
<td>0.05</td>
</tr>
<tr>
<td>21</td>
<td>Base preference bus</td>
<td>0.8</td>
<td>2</td>
</tr>
<tr>
<td>22</td>
<td>Base preference train</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>23</td>
<td>Access weight PT</td>
<td>0.085</td>
<td>2</td>
</tr>
<tr>
<td>24</td>
<td>Egress weight PT</td>
<td>0.165</td>
<td>1</td>
</tr>
</tbody>
</table>
With the suggested changes in parameter weights the transport mode division comes very close to the numbers extracted from the survey: 8% by PT, 49% by bike and 42% by car. Possibly these numbers could also be achieved with another combination of parameter weights.

Besides, it can be concluded that neighborhood roads are used too little for shortcuts according to the model results in comparison with BASEC, as some of these roads are far more used in reality then the model predicts (Table 7). Table 7 shows traffic volumes on sample locations to compare, in order to check the size and route choice of the population. The numbers are based on weekdays in both directions, and if not provided by BASEC, the separate directions are added up together. It can be seen that the total number of traffic movements has to be multiplied with a certain weight factor. It is suggested to multiply the outcome with ‘15’ to have on average the most realistic traffic volumes. Icons in Table 7 show whether or not the multiplied traffic volumes are in range of the real numbers (√), are below real numbers (x) or are above reality (x).

### Table 7 Traffic volume numbers for calibration on sample locations (extracted from [BASEC, 2015] and [Hermes, 2015])

<table>
<thead>
<tr>
<th>TYPE</th>
<th>STREET</th>
<th>BASEC 7-9h</th>
<th>BASEC 16-18h</th>
<th>MODEL 7-9h</th>
<th>MODEL 16-18h</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚌</td>
<td>BOSCHDIJK (peperstraat &gt; zoutstraat)</td>
<td>1,462</td>
<td>1,984</td>
<td>60 x 15 = 800 x</td>
<td>97 x 15 = 1,455 x</td>
</tr>
<tr>
<td>🚴</td>
<td>VELDMARSHALK MONTGOMERYLAAN (weverstraat &gt; looierstraat)</td>
<td>1,741</td>
<td>2,428</td>
<td>93 x 15 = 1,395 x</td>
<td>313 x 15 = 4,696 x</td>
</tr>
<tr>
<td>🚴</td>
<td>VESTDIJK (bleekstraat &gt; ten hagestraat)</td>
<td>1,904</td>
<td>1,707</td>
<td>189 x 15 = 2,835 x</td>
<td>168 x 15 = 2,520 x</td>
</tr>
<tr>
<td>🚴</td>
<td>MAURITSSSTRAAT (w dezwijgerstraat &gt; a van emmondstraat)</td>
<td>2,136</td>
<td>2,568</td>
<td>42 x 15 = 630 x</td>
<td>66 x 15 = 990 x</td>
</tr>
<tr>
<td>🚴</td>
<td>FREDERIKLAAN (lindenlaan &gt; berkenstraat)</td>
<td>1,010</td>
<td>1,474</td>
<td>6 x 15 = 90 x</td>
<td>20 x 15 = 300 x</td>
</tr>
<tr>
<td>🚴</td>
<td>TONGELREESTRAAT (w van konijnburgstraat &gt; valklaan)</td>
<td>1,116</td>
<td>1,736</td>
<td>79 x 15 = 1,185 √</td>
<td>104 x 15 = 1,560 √</td>
</tr>
<tr>
<td>🚴</td>
<td>VESTDIJK (bleekstraat &gt; ten hagestraat)</td>
<td>3,625</td>
<td>4,154</td>
<td>322 x 15 = 4,830 x</td>
<td>608 x 15 = 9,120 x</td>
</tr>
<tr>
<td>🚌</td>
<td>BOSCHDIJK (peperstraat &gt; zoutstraat)</td>
<td>732</td>
<td>918</td>
<td>46 x 15 = 690 x</td>
<td>40 x 15 = 600 x</td>
</tr>
<tr>
<td>🚌</td>
<td>VELDMARSHALK MONTGOMERYLAAN (weverstraat &gt; looierstraat)</td>
<td>117</td>
<td>249</td>
<td>25 x 15 = 375 x</td>
<td>107 x 15 = 1,605 x</td>
</tr>
<tr>
<td>🚴</td>
<td>MAURITSSSTRAAT (w dezwijgerstraat &gt; a van emmondstraat)</td>
<td>489</td>
<td>1,199</td>
<td>133 x 15 = 1,995 x</td>
<td>193 x 15 = 2,895 x</td>
</tr>
<tr>
<td>🚴</td>
<td>FREDERIKLAAN (lindenlaan &gt; berkenstraat)</td>
<td>4,327</td>
<td>3,617</td>
<td>620 x 15 = 9,300 x</td>
<td>376 x 15 = 5,640 x</td>
</tr>
</tbody>
</table>

### 6.2 ScenarIo Comparison

The base scenario of the current situation can be compared with the scenario as proposed in the municipal vision. Using a photo viewer on a computer can be helpful to distinguish differences between both map drawings, produced by the visualization application. Differences at two different time frames will be discussed in this section, namely the morning rush between 7h and 9h (Figures 26 and 27), and the evening rush between 16h and 18h (Figures 28 and 29). A uniform legend will be used in all produced maps. This legend is set inside the underlayer map (Figure 25). Traffic intensities of Public Transport are colored blue, while intensities of car and bike traffic are scaled from green to red.

**Figure 25 Legend of both underlayer maps (base scenario and municipal scenario)**
The municipal scenario will cause a calmer inner city at first sight between 7h and 9h according to the MSN-model. A large decrease of car traffic can be seen at the Hoogstraat, the Aalsterweg, the Leenderweg, the Tongelresestraat, the Kastanjelaan, the Mauritsstraat and the inner ring. However, at several spots the traffic intensity will increase as intended, like the canal area, the Philletelaan, the Karel de Grotelaan and the Fuultelaan, but also unintended at the Petrus Dondersstraat, the Hastelweg, the Willemstraat, the Nachttegaallaan, the Tramstraat and the roads between the Frederiklaan and the Hagenkampweg Noord. The traffic intensity possibly will be doubled on these roads because of cut-through traffic. However, the advantages of the proposal (no large traffic congestions on formerly critical roads) cannot be misunderstood.
The differences between both scenarios between 16h and 18h are similar to those between 7h and 9h, except that the evening rush is much more crowded with cars. There is even an increase of car traffic for the Leenderweg (as the Aalsterweg doesn't have a direct connection to the ring road in the municipal vision anymore). Also the Alberdingk Thijmlaan is more crowded to compensate the Aalsterweg, which was not the case in the morning rush. Another example is the Willemstraat, where the traffic intensity becomes quite critical, because of the closure of the inner ring. A conclusion could be that the municipal vision, on basis of adjustments in the inner city, does not lead to reduction of car traffic, as a lot of roads in the evening rush still suffer from traffic congestions. This can also be concluded from the statistical figures provided by the
visualization application (Table 8), where differences in transport mode division between the base scenario and the municipal scenario are very small. However, the proposals according to Public Transport were not applied in the model. Future research will have to show which impact High Quality Public Transport (‘HOV’) could have on the transport mode choice of individuals. Nevertheless, there are fewer cars inside the city center, which also was an objective of the municipal vision. The routes taken by car drivers through the inner city, for example along the Mauritsstraat, are less used in the municipal vision scenario at the expense of the new preferred routes (e.g. the Fuutlaan, the Philitelaan and the canal area). Beside these routes, two extra routes are going to be used by car drivers according to the MSN-model, being the Hastelweg (including lengthening to the Willemstraat) and the Alberdingk Thijmlaan. There is a risk these routes will suffer from traffic congestion.

Table 8 Statistical figures transport mode division different scenarios

<table>
<thead>
<tr>
<th>Time period</th>
<th>Base scenario</th>
<th>Municipal scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morning rush (7h – 9h)</td>
<td>Car 44%; Bike 44%; PT 10%</td>
<td>Car 44%; Bike 44%; PT 11%</td>
</tr>
<tr>
<td>Evening rush (16h – 18h)</td>
<td>Car 46%; Bike 47%; PT 5%</td>
<td>Car 47%; Bike 47%; PT 5%</td>
</tr>
<tr>
<td>Average of 24 hours</td>
<td>Car 42%; Bike 49%; PT 8%</td>
<td>Car 42%; Bike 49%; PT 8%</td>
</tr>
</tbody>
</table>

Yet, the question that needs to be answered is, whether or not the ring road is able to handle the traffic intensity, as the intensity will increase because of car drivers taking the ring road instead of the route through the city center in the municipal scenario. The ring road can be seen as ‘2x2 distributor road’, which has an capacity between 1,800 and 2,000 cars per lane per hour (Wegenwiki). As the ring road has 2 lanes and the time frame of the visualization application is 2 hours, the capacity will become to the utmost 8,000 car drivers in one direction. As the traffic intensity approaches the roads capacity the chance car drivers will encounter traffic congestion will grow fast starting from an I/C relation of 0.7 (Figure 30). Most ring road segments will have an intensity between 4,000 and 5,600 cars according to the model (I/C relation between 0.5 and 0.7), which is almost the same as the amount of ring road segments with an intensity between 5,600 and 7,200 cars (I/C between 0.7 and 0.9). However, at several points a traffic intensity between 7,200 and 9,600 cars is measured (I/C between 0.9 and 1.2), or at three spots even higher. These locations are divided over the ring road (between the crossing with the

![Figure 30 Intensity/Capacity relation](image)

(P0 is the percentage of all traffic over 24h that encounter traffic congestion) (RWS, 1999)

![Figure 31 Ring Road with visualized Intensity/Capacity](image)
Veldmaarschalk Montgomerylaan and the crossing with the Berenkuil - clockwise; and between the crossing with the Aalsterweg and the crossing with the Leenderweg), most of the time near crossings and locations where service roads start or end (Figure 31). In general the pressure on the ring road capacity increases under the municipal vision scenario. The connection between service roads and the ring road could be subject of future research, as there are more advanced intensity/capacity rules that can be applied in these merge situations.

Also bike and Public Transport visualization is done by means of the visualization application. Figures 32 and 33 show the comparison between the current situation and the proposed scenario between 7h and 9h. As the focus for bike traffic is at the inner city, the map is zoomed in at this particular area. However, the differences are minimal. As the route choice of bikers only depends on lowest cost (based on average speed on a certain road type and distance) the MSN-model does not take bike friendly ’historical’ routes into account, as proposed by the municipality, because the speed difference between the road types is minimal. Therefor conclusions about this proposal could not be drawn accurately. As shown by Table 8, also the effect on Public Transport cannot be predicted truthfully. Beside the fact that locations of Public Transport stops may be mapped a few hundred meters away from their real locations because of inaccurate coordinates translation from the Dutch coordinate system (RD-New) by TransCAD, also the proposals of High Quality Public Transport were not taken into account. In addition, it is not clear how individuals will respond to new proposed P+R locations. For now there are no clear differences between both scenarios.

6.3 CONCLUSION

To create a model which is equal to the real situation, transport mode division and traffic intensity numbers were collected. Based on the ‘Monitor Verkeer en Vervoer’ of the Eindhoven municipality the transport mode division should become 7% by PT, 49% by bike, 44% by car. By changing parameters inside the per_parameter.dat input file, in such way that Public Transport becomes less attractive and car more attractive comparing with the first run of the model, these goals were met closely. After that, the traffic intensity numbers were compared
with measurements extracted from BASEC, leading to a multiply factor for each individual of ‘15’ for all traffic, to create a good representation of reality. The municipal scenario can be compared now with the realistic base scenario. The proposals represented in the municipal scenario will lead to a calmer inner city, where new entry roads are being used as planned. However, some other roads inside the city center will be doubled in traffic, but will probably not cause new traffic congestions. Especially during the evening rush, which is more crowded than in the morning, two streets (the Hastelweg till the Willemstraat and the Alberdingk Thijmlaan) need extra attention.

At first sight the suggestions of the municipal vision will not lead to a change in transport mode choice. However, proposals for a new more frequent High Quality Public Transport in combination with new P+R locations were not taken into account, as it is quite time consuming and hard to implement in the MSN-model. For bikers only changes in road types were applied. Apparently, these changes do not stimulate bikers using historical routes. It is not sure bikers prefer longer car free routes over shorter car streets, which can be subject for further research. There can be chosen to make car streets even less attractive for bikers. By including more parameters, the MSN-model may predict slow mode preferences more accurately.

Most of the cars that crossed the city center before will now end up on the ring road, as a result of the municipal vision. The question is whether or not the ring road can handle the increased amount of traffic. Some parts of the ring road will be so crowded that 30% of all cars driving on that road section will encounter traffic congestion. However, because of fewer traffic intersections with the ring road, the amount of traffic can be handled in a smoother motion than in the current situation. Attention should also be paid to the way service roads are connected to the ring road in order to make the traffic flow as efficient as possible, as the interaction between those lanes can influence the capacity in a negative way.
7. CONCLUSIONS, RECOMMENDATIONS & DISCUSSION

7.1 CONCLUSIONS

In this project different activity based models were studied, as well as the municipal vision for 2040. The municipal vision focuses mainly on the inner city area, by proposing fewer connections to the ring road, car free inner ring and lower speed on most roads, in order to reduce car traffic in the inner city and provide an alternative for this car traffic by a new ring road structure of service roads. The recently developed MSN-model was applied to the Eindhoven area to examine the municipal vision. This model first applied to the Rotterdam region, was recalibrated for the Eindhoven region and input files were generated. The current situation is simulated as the base scenario and compared with a scenario based on the municipal vision. This municipal scenario includes all proposed changes for car traffic. Several scripts were designed for the input files to expedite the work for both scenarios. In order to calibrate the current situation, traffic intensity numbers provided by BASEC and transport mode division figures of the ‘Monitor Verkeer en Vervoer’ were used, which is a residents survey of Eindhoven.

Working with this model has led to new experiences, on the basis of which improvements to the model can be suggested. These improvements are mentioned in the recommendation section. Because the MSN-model did not contain a visual component yet, an application was developed to analyze the model’s textual output for a prompted time frame. The application also forms a bridge between the analyzed data and TransCAD – a geographic information system – to visualize the output on a prepared TransCAD map. The visualization application consists of three components. The first, the analyze component, reads and categorizes MSN’s output data by transport mode. The third component is able to project the categorized data on a TransCAD underlayer map, while the second component connects both components to each other. The components are written in C++, Visual Basic and GISDK, as these scripting languages have specific possibilities. TransCAD only can understand GISDK, and Visual Basic is one of the only languages that can call GISDK from outside the TransCAD environment. The analyze component can easily be integrated inside the MSN-model as both codes are made in C++ language.

The differences between both scenarios (the current state and the future situation according to the municipal vision) will be discussed taking the research questions in consideration.

*Will the municipal policy lead to less car traffic for the region in general, less car traffic inside the inner city area, and an increase of bike traffic and public transport inside this area?*

The municipal policy will not lead to less car traffic in general, however it should be stated that proposals for Public Transport and P+R locations were not taken into account and could have led to less car traffic. For now the result is that the percentage of car traffic remains the same, so the policy will also not lead to more car traffic. The inner city area will encounter less car traffic and congested streets will become less congested. Two streets, the Hastelweg till the Willemstraat and the Alberdingk Thijmlaan, where currently is no congestion, however, will be much increased in traffic during the evening rush and therefore need extra attention. For now the bike traffic and public transport do not increase nor decrease inside the inner city area, however, not every suggestion of the vision could be applied.
**Will the policy solve existing bottlenecks inside the inner city area, and ensure that no new bottlenecks will arise, meaning car drivers will not search for new shortcuts?**

The policy solves existing bottlenecks inside the inner city area, like the Mauritsstraat and the inner ring, and direct car traffic to the ring road. Despite of the traffic intensity in the Alberdingk Thijmlaan increases as a shortcut for the restricted Aalsterweg, it will probably not lead to a new bottleneck, because of the wide street profile and the good connection to service roads on both sides of the ring road. However, the street profile of the Hastelweg is smaller, and the length of the congestion is predicted over a much longer distance. New research should be done to examine either these streets should be closed for cars, or the street structures should be adapted to the increased car traffic in this street. Besides, some roads will double in traffic intensity but will not lead to new congestions. These routes are mainly taken by individuals with their destination inside the inner city area. When P+R locations and High Quality Public Transport are taken into account, this problem may be leveled out.

**Can the ring road handle a larger traffic intensity with regard to the proposed system of through- and local traffic, compared to the capacity according to the handbooks?**

The capacity of the ring road according to handbooks is between 7,200 and 8,000 cars for 2 lanes and a 2 hour time frame. Most road sections of the ring road will have traffic intensities up to 7,400 cars. However, some other parts of the ring road will be so crowded that 30% of all cars driving on that road section will encounter traffic congestion. Because of fewer crossings the amount of traffic can be handled in a smoother motion than in the current situation, but still attention should be paid to the way service roads are connected to the ring road in order to make the traffic flow as efficient as possible, as the interaction between those lanes can influence the capacity in a negative way. The next step is to direct people from the ring road towards the highway, in case of though traffic towards Helmond for example. Also the use of P+R locations, as proposed, can have positive effects on the traffic intensity on the ring road. Problems in traffic intensity for the ring road should be addressed, in order to prevent that car drivers will drive through the inner city again, in an attempt to avoid traffic congestion.

**How to extend the MSN-model in order to visualize the model's output?**

The MSN-model had to be extended with a tool to analyze the model's output. To make this tool as efficient as possible, the output structure is rearranged in two lines for each individual: one line for all locations of an individual and one line for the corresponding times. The analyze component of the visualization tool searches for times that fit between the start time and end time the user is interested in. For these times the transport mode and location node is extracted and saved separately within the visualization tool. When the searching is done, all saved locations are categorized by transport mode, and thereafter summed up when the locations are similar to each other. For each transport mode a new output file is created. This output file is connected to TransCAD by a script, to update the traffic intensities and to plot several maps. The produced maps show the traffic intensities for each road node and Public Transport stop. The use of road segments instead of road nodes was not possible, because then TransCAD overruled the road category structure of displayed line thickness. Summarized, for each entered time frame a map can be produced. It is possible to prompt the situation for several consecutive times or time frames to visualize the movements over a day. However, as the connection between the visualization tool and TransCAD requires user interaction, it can become a rather time consuming process, depending on the number of frames.
7.2 RECOMMENDATIONS

In this section two types of recommendations will be made. First all recommendations will be noted concerning future research that is recommended to conduct, thereafter recommendations will be discussed to improve the MSN-model and the visualization tool. Strong recommendations towards policymakers are not possible yet, given the exploratory stage the research is in. Nevertheless, it is advised to execute the proposed plans for 2040 based on the first results.

Future research
The capacity of the ring road nearby service road access ramps and service road exit points should be investigated, in order to develop plans to guarantee smooth flows. Also there should be paid extra attention to roads that attract more car traffic than their capacity allows.

Future research should also be done to access the likely effects of High Quality Public Transport and P+R facilities on traffic intensity and transport mode division. It is possible to implement this using the current MSN-model, however, it is time consuming to develop and digitize a whole new Public Transport service plan, detailed for a whole day.

MSN-model improvements
In order to have results from the supernetwork model as quickly as possible, it is very much recommended that the model can detect errors in input files, before the MSN-model starts running. Some tests need to be developed to check the input files on data structure errors.

Another recommended extension of the MSN-model would be the interaction between household members. At the moment car passengers are not taken into account, which is solved by attaching cars to more members of the household. It should be possible to be a car passenger, so household members or friends going to the same activity could travel by the same car, which will influence the traffic intensity as well (however, the multiply factor will be higher then). Also, the MSN-model does not take positions of other individuals into account while planning routes, to make it possible to predict traffic congestions by the model. This could also be a welcome extension of the model, where individuals could choose another route, when there are too many cars on the same road section. A maximum amount of cars for each road section could be set per minute, represented by ‘keys’, where without key the weight for the road section will increase. In this way also the time being in a traffic jam could be represented, as currently the cars do not lose time being in a traffic congestion. Besides, traffic lights could be introduced, which also are time consuming elements, represented by a new road category with high weights, in front of crossings in the research area. Actually, current available traffic light systems could be simulated. Besides, the MSN-model does not cover pedestrian output, and also the new visualization application is not able to handle pedestrians yet. The difficulty is the way how to visualize pedestrian flows between start point and destination in TransCAD, because they don’t stick the road network, but use informal options in their route choice.

Visualization improvements
Furthermore, the individuals recorded in the activity program input files were attached to a four digit postcode extracted from MON/OVIN data. A script divides those individuals randomly across the six digit postcode areas in the four digit postcode area (represented by one node in
the road network). Therefor a lot of individuals are located on the same node (center of the six digit postcode area), which can cause traffic congestion towards a main road. Individuals could also be attached to a random node within the postcode area. Also the neighborhood statistics could play a role in dividing individuals over the area.

7.3 Discussion
The following choices and assumptions were made during the preparation of the input files of the MSN-model, which influenced the presented final results. Conclusions extracted from the results should also take these choices into account.

Dimension Road Network
The city of Eindhoven is fully detailed, while all cities in the surrounding have been represented by a node, connected by regional roads and highways, and therefore could cause some impurities. Because of some missing roads, the road where traffic enters the city could be different and especially the connection between the road network and the Public Transport network, outside the Eindhoven area, could be unrepresentative.

Dimension Public Transport
The Public Transport within Eindhoven is fully detailed, as well as bus routes of Connexxion and Arriva towards nearby villages. Besides, all train connections are part of the Public Transport network. However, in areas outside Eindhoven, where bus connections were not covered, people who don’t have a car have to walk to the nearest train station, leading to very long travel times. In order to deal with this, all forced walking distances were reduced, however this could also lead to a more positive connection to Public Transport, when there is no proper connection to the train network in reality as well, due to lack of buses in this area.

Population Composition
People who travel to the Eindhoven region to conduct an activity are taken into account, as well as people who have their home location inside the Eindhoven region. Other people conducting activities outside the Eindhoven region, but travel through Eindhoven, will probably take the highway or train and do not affect the traffic situation within the city centre of Eindhoven.

The population size of the Eindhoven case exists of around 12,000 individuals, extracted from ten different years of MON/OVIN data, as the population of just one database was not big enough to handle detailed traffic analyses. Therefor the used data could be obsolete. The home location individuals are connected to, are based on the four digit postcode location, but randomly transformed into a six digit postcode location. Besides, a personal weight factor – provided by MON/OVIN – is applied, to create a representative composition of the population. The factor is implemented before attaching the individual to the six digit home location node, to vary the randomly picked home location. However, as it is random it does not take neighborhood statistics into account.

Besides, the second multiply factor that is used to scale up the population during the calibration phase, is now based on specific sample locations. These locations could have been different, possibly resulting in another factor, which subsequently could lead to another conclusion about the road intensities. The same applies to the external parameter weights to calibrate the MSN-model.
REFERENCES


IBM Corp. (2013). IBM SPSS Statistics for Windows (Version 22) [Software]. Armonk, NY.


Microsoft Corp. (2010). Microsoft Visual C++ 2010 Express (Version 10.0) [Software]. Redmond, WA.


Python Software Foundation. (2012). WinPython (Version 2.7.6.3) [Software]. Wilmington, DE.


APPENDIX

1. INSERTING ACTIVITY EPISODES INTO PROJECT AGENDAS (Miller & Roorda, 2003)

Case 1 – The new episode being added to the project agenda is added within an existing episode, thereby splitting it into two episodes. This type of insertion is only allowed in the work project agenda, whereas the episode being split is the primary work event.

Case 2 – The new episode being added to the project agenda can be inserted between two existing episodes, but the new episode only overlaps one of the episodes because the new episodes’ end (or start) time coincides with one of the existing episodes’ end (or start) time.

Case 3 – The new episode being added can be inserted between two existing episodes. However, the start and end times of the new episode do not coincide with either of the existing start or end times so the new episode overlaps both existing episodes.

Case 4 – A long duration activity might completely overlap one or more shorter episodes. This case is disallowed. In the current model, the new long duration activity episode would be rejected from the project agenda.

2. INSERTING AN ACTIVITY INTO A PERSON SCHEDULE (Miller & Roorda, 2003)

4. Districts and Radials (Helms, 2013)
5. **RESULTS OF THE SLOW MOTION, FAST FORWARD PRINCIPLE (HELMS, 2013)**

6. **HIGH QUALITY PUBLIC TRANSPORT (HELMS, 2013)**
7. **Python script to reduce amount of land-use locations**

*By combining same activity-types within the same six digit postcode area*

```python
import os
import numpy as np
import pandas as pd

mapp = r'D:\Documents\Master\Afstuderen\7MM37'
datain = os.path.join(mapp, 'transcad and new scenario', 'New data', 'land_use.txt')
dataout = os.path.join(mapp, 'transcad and new scenario', 'New data', 'land_use_reduce.dat')

data = pd.read_csv(datain, sep=',')
data = data.groupby(['D', 'A', 'B', 'C', 'F', 'G', 'H', 'I', 'J', 'K', 'L'])['E'].sum().reset_index()
data = data.sort_index(axis=1)
data.L = pd.factorize(data.L, sort=False)[0]
data.to_csv(dataout, sep='\t', header=None, index=False)
print 'Done'
```

8. **Python script to create all activity program files**

*Activities extracted from MON/OVIN, making use of postcode X,Y coordinates and land-use*

```python
import os
import numpy as np
import pandas as pd
import gc
import sys, time
import fileinput
import glob
try:
    from IPython.display import clear_output
    have_ipython = True
except ImportError:
    have_ipython = False

class ProgressBar:
    def __init__(self, iterations):
        self.iterations = iterations
        self.prog_bar = '['
        self.fill_char = '*'
        self.width = 40
        self._update_amount(0)
    
    def animate_ipython(self, iter):
        if sys.platform.lower().startswith( 'win' )
            print self, '',
        else:
            print self, chr( 27 ) + '\[A'
        self.update_iteration( iter )

    def animate_noipython( self, iter ):
        self.update_iteration( iter )

    def update_iteration( self, iter ):
        if iter:
            self.prog_bar += self.fill_char
        else:
            self.prog_bar = '['
        self.prog_bar = self.prog_bar[:self.width]
        self.prog_bar += ' ' * (self.width - len(self.prog_bar))
        sys.stdout.flush()

    def _update_amount(self, amount):
        if amount:
            self.prog_bar += self.fill_char
        else:
            self.prog_bar = '['
        self.prog_bar = self.prog_bar[:self.width]
        self.prog_bar += ' ' * (self.width - len(self.prog_bar))
        sys.stdout.flush()

def _init__(self, iterations):
    self.iterations = iterations
    self.prog_bar = '['
    self.fill_char = '*'
    self.width = 40
    self._update_amount(0)

def animate_noipython( self, iter ):
    if sys.platform.lower().startswith( 'win' )
        print self, '',
    else:
        print self, chr( 27 ) + '\[A'
    self.update_iteration( iter )

def animate_ipython(self, iter):
```
print '\r', self,
sys.stdout.flush()
self.update_iteration(iter + 1)

def update_iteration(self, elapsed_iter):
    self._update_amount((elapsed_iter / float(self.iterations)) * 100.0)
    self.prog_bar += ' %d of %s complete' % (elapsed_iter, self.iterations)

def _update_amount(self, new_amount):
    percent_done = int(round((new_amount / 100.0) * 100.0))
    all_full = self.width - 2
    num_hashes = int(round((percent_done / 100.0) * all_full))
    self.prog_bar = [' ' * self.fill_char * num_hashes + ' ' * (all_full - num_hashes) + '']
    pct_place = (len(self.prog_bar) // 2) - len(str(percent_done))
    pct_string = '%d%%' % percent_done
    self.prog_bar = self.prog_bar[0:pct_place] + (pct_string + self.prog_bar[pct_place + len(pct_string):])

def __str__(self):
    return str(self.prog_bar)

def superfunc(mapp, year, OVIN, factordev):
    if(OVIN==False):
        filename = os.path.join(mapp,'input data raw','MON-OVIN','Databestand MON '+str(year)+'.csv')
    else:
        filename = os.path.join(mapp,'input data raw','MON-OVIN','OVIN'+str(year)+'_Databestand.csv')

    postfn = os.path.join(mapp,'transcad and new scenario','New data','land_use.dat')
    pc4fn = os.path.join(mapp,'transcad and new scenario','New data','pc4.txt')
    savefn = os.path.join(mapp,'transcad and new scenario','New data','Act_Prog','act_prog'+str(year)+'.d1')
    savefn2 = os.path.join(mapp,'transcad and new scenario','New data','Act_Prog','per_profile'+str(year)+'.d2')
    savefn3 = os.path.join(mapp,'transcad and new scenario','New data','Act_Prog','per_home'+str(year)+'.d3')

    tmp = pd.read_csv(filename, sep=';')
    renameOVIN = {'Doel':'aankbzh','ActDuur':'aktduur','HHBestInk':'inkomen',
                  'Opleiding':'opleid','BetWerk':'betaaldw','HHAuto':'hauto',
                  'HHFiets':'hfiets','Rijbewijs':'rijbewij','HoofdAuto':'hfdauto','OPID':'persid',
                  'Rvm':'kvv','RVm':'kvv','OVGebruik':'persopv','WoGem':'wogem'}

    tmp = tmp.rename(columns=renameOVIN)
    if(OVIN==True):
        try:
            tmp['hhid'] = tmp['persid']
        except:
            pass

    import re, string;
    pattern = re.compile('[\W_]+')

    def an_strip(j):
        return pattern.sub('', j)

    orgnames = tmp.columns.values
    newnames = map(an_strip,tmp.columns.values)
    newnames = map(string.lower, newnames)

    colomdict = {orgnames[i]:newnames[i] for i in xrange(len(orgnames))}

    tmp = tmp.rename(columns=colomdict)
tmp = tmp[tmp['vertrekp']==1]
if(year==2004):
    nrlst = [772,866,794,861,820,1771,753,858,848]
    tmp = tmp[tmp['aankgem'].isin(nrlst)|tmp['vertgem'].isin(nrlst)]
else:
    nrlst = "772|866|794|861|820|1771|753|858|848"
    tmp = tmp[tmp['aankgem'].str.contains(nrlst)|tmp['vertgem'].str.contains(nrlst)]
if(year==2004):
    allowed = \[2,3,4,5,6,7,8,9,10,11,12,13,14,15\]
    tmp = tmp[tmp['aankbzh'].isin(allowed)]
else:
    allowed = "2|3|4|5|6|7|8|9|10|11|12|13|14|15"
    tmp = tmp[tmp['aankbzh'].str.contains(allowed)]

for j in todrop:
    tmp = tmp.dropna(subset = [j])

for j in todrop:
    return str(row['hhid'])+str(row['leeftijd'])

for j in todrop:
    return str(row['jaar'])+'/'+str(row['maand'])+'/'+str(row['dag'])+'_'+str(row['persid'])

for i, group in enumerate(tmp.groupby('uniq')):
    factor = int(group[1]['factorp'].max())
    factor = max([1,int(factor/factordev)])
    toadd = pd.concat([group[1]]*factor)
    toadd['itjes'] = np.ones(len(toadd),dtype=int)
grouplen = len(group[1])
for ii in xrange(factor):
    toadd[‘itjes’][ii*grouplen:(ii+1)*grouplen] *= ii
    toadd[‘uniqnew’] = toadd.apply(lambda row: func3(row), axis=1)
addlist.append(toadd)
p.animate(i)

print
newtmp = pd.concat(addlist)

print ‘persid update ..’,
tmp = newtmp
def func1(row):
    return str(row[‘persid’])+str(row[‘itjes’])
tmp[‘persid’] = tmp.apply(lambda row: func1(row), axis=1)
print ‘done’

print ‘hhid update ..’,
def func2(row):
    return str(row[‘hhid’])+str(row[‘itjes’])
tmp[‘hhid’] = tmp.apply(lambda row: func2(row), axis=1)
print ‘done’

tmp = tmp.sort([‘uniqnew’], ascending=[True])
tmp = tmp.reset_index()

postdata = pd.read_csv(postfn,sep=‘\t’,header=None, skiprows=9,usecols=[0,1,2,3,8])

c4data = pd.read_csv(pc4fn,sep=‘,’,header=None)

recoder = {2:np.random.choice([1,3],1)[0], 3:4, 4:np.random.choice([1,3],1)[0], 5:np.random.choice([1,3],1)[0], 6:np.random.choice([1,3],1)[0], 7:7, 8:2, 9:6, 10:np.random.choice([1,3],1)[0], 11:8, 12:np.random.choice([1,3],1)[0], 13:5, 14:np.random.choice([1,3],1)[0], 15:np.random.choice([1,3],1)[0]}

lst = []
lst2 = []
lst3 = []

p = ProgressBar(len(tmp[‘uniqnew’].unique())-1)

for i,j in enumerate(tmp.groupby(‘uniqnew’):
    j = j[1]

    # LST2
    foundpst = int(j[‘vertpc’].values[0])
    foundland = postdata[(postdata[8] == int(foundpst))].reset_index()[‘index’].values

    if len(foundland) > 0: #this part search for 6 digit postcode data in landuse.dat ...
        result = np.random.choice(foundland,1)[0]
        ingw = int(postdata.iloc[[result]][‘index’].values[0][0])
        ingwx = int(postdata.iloc[[result]][‘index’].values[0][1])
        ingwy = int(postdata.iloc[[result]][‘index’].values[0][2])
    else: # ... if not founded 4 digit postcode data from pc4.txt will be used.
        not_done = True
        while not_done == True:
            try:
                ingw = int(pc4data[pc4data[0] == foundpst].values[0][1])
                ingwx = int(pc4data[pc4data[0] == foundpst].values[0][2])
                ingwy = int(pc4data[pc4data[0] == foundpst].values[0][3])
not_done = False
def exception, e:
    print e
    print foundpst
    foundpst = foundpst + 1

strr2 = [j['persid'].values[0], j['hhid'].values[0], j['geslacht'].values[0], j['leeftijd'].values[0], j['inkomen'].values[0], j['opleid'].values[0], 1, j['betaaldw'].values[0] == 3 else 0, 0, 1 if j['hauto'].values[0] is 1 and int(j['hdauto'].values[0]) == 1 or (int(j['hauto'].values[0]) > 1 and (int(j['kvv'].values[0] < 3))) else 0, 0, 1 if int(j['hfiets'].values[0] > 0 else 0, 0, 1 if int(j['persopv'].values[0] == 1 else 0, 0, j['kvv'].values[0], j['rijbewij'].values[0], 1 if int(j['wogem'].values[0] == 772 else 0, 0]]

strr2 = [str(k) for k in strr2]
strr2 = '\t'.join(strr2)
lst2.append(strr2)

# LST3
strr3 = [ingw, ingwx, ingwy, j['vertpc'].values[0], 1, year, 0, 0, 0]
strr3 = [str(k) for k in strr3]
strr3 = '\t'.join(strr3)
lst3.append(strr3)

# LST1
aantal_ev = j['uniqnew'].count()

strr_p = [0, j['persid'].values[0], aantal_ev, -1, int(j['vertuur'].values[0])*60+int(j['vertmin'].values[0]), -1, 0, 0, 0]

strr_p = [str(k) for k in strr_p]
strr_p = '\t'.join(strr_p)
lst.append(strr_p)

for jj in xrange(aantal_ev):
    aanres = -1
    aankbzh_jj = recoder[int(j['aankbzh'].values[jj])]
    if aankbzh_jj in [1, 3, 7, 8]:
        postalcode_jj = j['aankpc'].values[jj]
        found_indices = postdata[(postdata[0] == aankbzh_jj) & (postdata[8] == int(postalcode_jj))].reset_index()['index'].values
        if len(found_indices) > 0:
            aanres = np.random.choice(found_indices, 1)[0]
        else:
            found_indices = postdata[(postdata[0] == aankbzh_jj) & (postdata[8] == int(postalcode_jj))].reset_index()['index'].values
            if len(found_indices) > 0:
                aanres = np.random.choice(found_indices, 1)[0]
            else:
                print 'Error! No fixed land_use location suitable. Make sure every activity type has at least one location!'
    strr = [aankbzh_jj,
            aanres,
            j['aktduur'].values[jj],
            np.binary_repr(jj),
            np.binary_repr(jj+2),
            0, 0, 0, 0]
```python
strr = [str(k) for k in strr]
strr = '\t'.join(strr)
lst.append(strr)
p.animate(i)

with open(savefn, "w") as f:
    f.writelines("\n'.join(lst))
f.writelines("\n")

with open(savefn2, "w") as f:
    f.writelines("\n'.join(lst2))
f.writelines("\n")

with open(savefn3, "w") as f:
    f.writelines("\n'.join(lst3))
f.writelines("\n")

gc.collect()

if __name__ == '__main__':
    mapp = r'D:\Documents\Master\Afstuderen\7MM37'
factordev = 10000.    # Divide personal MON/OVIN factor by number; 10000 = not apply factor.
years = np.arange(2004,2014,1)
OVINs = np.where(years>=2010,True,False)

print years,OVIN,factordev
print
for i in xrange(len(years)):
    year = years[i]
    OVIN = OVINs[i]
    print '\t',year,OVIN
    print
    superfunc(mapp, year, OVIN, factordev)

mapp = r'D:\Documents\Master\Afstuderen\7MM37'
output1 = os.path.join(mapp,'transcad and new scenario','New data','act_prog.dat')
output2 = os.path.join(mapp,'transcad and new scenario','New data','per_profile.dat')
output3 = os.path.join(mapp,'transcad and new scenario','New data','per_home.dat')

os.chdir('D:\Documents\Master\Afstuderen\7MM37\transcad and new scenario\New data\Act_Prog')

file_list = glob.glob("*.d1")
with open(output1, 'w') as file:
    input_lines = fileinput.input(file_list)
    file.writelines(input_lines)

file_list = glob.glob("*.d2")
with open(output2, 'w') as file:
    input_lines = fileinput.input(file_list)
    file.writelines(input_lines)

file_list = glob.glob("*.d3")
with open(output3, 'w') as file:
    input_lines = fileinput.input(file_list)
    file.writelines(input_lines)
```

This code appears to be a script for handling data, possibly related to urban planning or a similar field. It includes functions for reading and writing files, handling lists and strings, and a main block with calculations and file operations. The script seems to be parsing and manipulating data for different years, applying factors, and writing the results to new files in a specific directory structure.
9. C++ script to analyze the MSN-output and prepare TransCAD readable files

```cpp
#include "stdafx.h"
#include <algorithm>
using namespace std;

int main()
{
    clock_t start, end;
    start = clock();

    cout << "Please wait..... model is creating space for variables. ";
    #include "declare.h"
    cout << "Done!\n\n";

    //load file
    loop:
    cout << "Enter file name:\n";
    cin >> fname;
    file.open(fname.c_str());
    if (!file)
    {
        cout << "File not found! Make sure the filename does not consist any whitespaces!\n\n";
        cin.clear();
        cin.ignore(INT_MAX, '\n');
        goto loop;
    }
    cout << "File accepted\n\n";

    //fastmode activation
    if (!fastmode) {cout << "Debug-mode on\n";}
    if (fastmode) {cout << "Debug-mode off\n";}

    //overview activation
    if (overview) {cout << "Overview-file on\n";}
    if (!overview) {cout << "Overview-file off\n";}

    //excel activation
    if (excel) {cout << "Execute Excel on\n";}
    if (!excel) {cout << "Execute Excel off\n";}

    //timeframe activation
    if (timeframe) {cout << "Timeframe on\n\n";}
    if (!timeframe) {cout << "Timeframe off\n\n";}

    //enter time window
    loop1:
    cin.clear();
    cin.ignore(INT_MAX, '\n');
    cout << "Enter research time by [hours] [space/enter] [minutes]:\n";
    cin >> hours;
    if(hours < 0 || hours > 23) {goto invalid;}
    cin >> minutes;
    if(minutes < 0 || minutes > 59) {goto invalid;}
    time = hours * 60 + minutes;
    if (time > 10 && time < 1440)
    {
        cout << "Valid input. Time in minutes: " << time << "\n\n";
        if (timeframe) {goto loop2;}
        if (!timeframe) {goto loop3;}
    }
    else
    {
        invalid:
        cout << "Invalid input! You have to enter a time between [0:10] and [24:00].\n\n";
    }
```
goto loop1;
}

loop2:
cin.clear();
cin.ignore(INT_MAX, '\n');
cout << "Enter second research time by [hours] [space/enter] [minutes]:\n";
cin >> hours2;
if(hours2 < 0 || hours2 > 23) {goto invalid2;}
cin >> minutes2;
if(minutes2 < 0 || minutes2 > 59) {goto invalid2;}
time2 = hours2 * 60 + minutes2;
if (time2 > 10 && time2 < 1440 && time2 > time) {
    cout << "Valid input. Second time in minutes: " << time2 << "\n";
go to loop3;
} else {
    invalid2:
    cout << "Invalid input! You have to enter a time between [" << hours << ":" << minutes << "] and [24:00]\n"
    goto loop2;
}

loop3:
if(fastmode) {cout << "Please wait.....\n";}
//reading file amount of lines
wordcounter = 0;
if(file.is_open()) {
    while(file.good())
    {
        count = 0;
        getline(file, line);
        if(!fastmode) {cout << "reading new line: ";}
        for (int k=0; k<int(line.length()); k++)
        {
            if (isspace(line[k])) {count++; wordcounter++;}
        }
        if(!fastmode) {cout << count << " words\n";}
        if(count > maxcount) {maxcount = count;}
        countArray[r] = count;
        r++;
    }
    if(count != 0)
    {
        cout << \nThe last line is missing in the input file. Extra line
created.\n";
        cout << "---\n";
        cout << "Do not edit the input file manually! Be aware that a whitespace
at the end of each line is required, otherwise the output of this program is not useful
anymore!\n";
        cout << "---\n";
        r++;
    }
    file.close();
} else {
    cout << \nThere is something wrong with the imported file...\n";
    system("pause>nul");
go to end2;
}

cout << \n" << r << " lines in file\n";
cout << wordcounter << " words in file\n";
if(!(fastmode || count!=0))
{
    cout << "\nGlobal reading finished. Press key to continue detailed reading...\n\n";
    system("pause\nul");
}
if(fastmode)
{
    cout << "\nPlease wait.....\n\n";
}
//reading file words on line
file.open(fname.c_str());
if(file.is_open())
{
    for(int l = 0; l < r; ++l)
    {
        if(!fastmode){cout << "\nCurrent working line: " << (l+1) << " (" << countArray[l] << " words)\n";}
        //spliting up variables of odd lines and even lines
        if (oddeven == 2) {oddeven = 1;}
        if (oddeven == 3) {oddeven = 0;}
        if (oddeven == 0)
        {
            l1++;
            if(!fastmode){cout << "reading line.. location\n";}
            for(int i = 0; i < (countArray[l]); ++i)
            {
                ci++;        
                file >> analysetimeArray[cj];
                //if(!fastmode) {cout << "writing time nr: " << cj << " with value: " << analysetimeArray[cj] << "\n";}
                if(j==0) {timeArray[l2] = cj;};
                if((i+1)==countArray[l])
                {
                    if(i==0) {locationArray[l1] = ci;}
                    if(i+1==countArray[l])
                    {
                        if(i==0)
                        {
                            if(!couter)
                            {
                                if(!fastmode) {cout << "---\nNote: Your input file has empty lines! Press key to continue...\n---\n\n"; system("pause\nul"));
                                if(fastmode) {cout << "---\nNote: Your input file has empty lines!\n---\n\nPlease wait.....\n\n";}
                                couter = true;
                            }
                            ci++; //create empty position at last to keep number of time equal to number of location
                        }
                    }
                }
        if (oddeven == 1)
        {
            l2++;
            if(!fastmode){cout << "reading line.. time\n";}
            for(int j = 0; j < (countArray[l]); ++j)
            {
                cj++;        
                file >> analysetimeArray[cj];
                //if(!fastmode) {cout << "writing time nr: " << cj << " with value: " << analysetimeArray[cj] << "\n";}
                if(j==0) {timeArray[l2] = cj;}
            }
        }
    }
}
58
oddeven = 3;

} else
{
    cout << "\nThere is something wrong with the imported file...\n\n";
    system("pause\nul");
    goto end2;
}

if(!fastmode){cout << "\nDetailed reading finished. Press key to continue analyses...\n\n"; system("pause\nul");}

//test variables on equal transport modes
for(int t = 0; t < ((r-1)/2); ++t)
{
    if(analyselocationArray[locationArray[t]]!=analysetimeArray[timeArray[t]])
    {
        cout << "\nInvalid File! The data in the imported file is not consistent in transport mode!\n";
        cout << analyselocationArray[locationArray[t]] << " is not " << analysetimeArray[timeArray[t]] << " for positions " << (((t+1)*2)-1) << " and " << (((t+1)*2) << "\n";
        goto end;
    }
    else
    {
        if(!fastmode){cout << analyselocationArray[locationArray[t]] << " is " << analysetimeArray[timeArray[t]] << " for positions " << (((t+1)*2)-1) << " and " << (((t+1)*2) << "\n";}
        //used for statistics
        P = stoi(analyselocationArray[locationArray[t]].c_str());
        if (P==0) {pcar++; ptotal++;}
        if (P==1) {pbike++; ptotal++;}
        if (P==2) {ppt++; ptotal++;}
    }
}
if(!fastmode){cout << "\nValid File. Press key to continue analyses...\n\n";}
if(!fastmode){system("pause\nul");}

//analyses
for(int tr = 0; tr <= cj; ++tr)
{
    //transform string to double in order to compare with time
    analtimeArray[tr] = atof(analysetimeArray[tr].c_str());
}

backin:
    cout << "Number of time values for analyses: " << cj << "\n\n";
    for(int m = 0; m <= cj; ++m)
    {
        if(!timeframe)
        {
            if(time >= analtimeArray[m] && time < analtimeArray[m+1] && analtimeArray[m] > 10)
            {
                if(!fastmode){cout << "Match! Between " << analtimeArray[m] << " and " << analtimeArray[m+1] << "\n";}
                resolveArray[num] = m;
                num++;
            }
        }
        else
        {
            for(int x = 0; x < maxcount; ++x)
if (time <= analtimeArray[m] && time2 > analtimeArray[m+1+x] &&
    analtimeArray[m] > 10 && analtimeArray[m+1+x] > 10)
{
    if(analtimeArray[m+x] == analtimeArray[m+1+x])
    {
        if(!fastmode){cout << "Match dropped because of equality of result " << analtimeArray[m+x] << " = " << analtimeArray[m+1+x] << "\n";}
        err++;
    } else
    {
        if(!fastmode){cout << "Match! Between " << analtimeArray[m+x] << " and " << analtimeArray[m+x+1] << "\n";}
        resolveArray[num] = m+x;
        num++;
    }
} else
{
    m = m + x;
    x = maxcount;
}
}
if(!timeframe) {cout << num << " matches for the given time: " << time << "\n";}
if(timeframe) {cout << err << " matches dropped because of equality of result\n";}
if(timeframe) {cout << num << " matches between the given times: " << time << " and " << time2 << "\n";}
if(num == 0) {goto again;}
if(!fastmode) {cout << "Press key to show results...\n"; system("pause>nul");}

//results
cout << "\nPlease wait..... model is preparing save of several files:";
if(overview) {
    if(!timeframe) {oo << "results/overview" << time << "_" << fname;}
    if(timeframe) {oo << "results/overview" << time2 << "_" << fname;}
    outname = oo.str();
    output.open(outname);
}
for(int rlt = 0; rlt < num; ++rlt)
{
    for(int mt = 0; mt <= l2; ++mt)
    {
        if(resolveArray[rlt] > timeArray[mt] && resolveArray[rlt] < timeArray[mt+1])
        {
            force:
            if(!fastmode){cout << "transportmode: " << analysetimeArray[timeArray[mt]] << " result: " << analyselocationArray[resolveArray[rlt]] << "\n";}
            if(overview) {output << analysetimeArray[timeArray[mt]] << "," << analyselocationArray[resolveArray[rlt]] << "\n";}
            //create separate data output files based on transportmode
            if(analtimeArray[timeArray[mt]] == 0) {carArray[carcounter] = analyselocationArray[resolveArray[rlt]];
                carcounter++;}
            if(analtimeArray[timeArray[mt]] == 1) {bikeArray[bikecounter] = analyselocationArray[resolveArray[rlt]];
                bikecounter++;}
            if(analtimeArray[timeArray[mt]] == 2) {ptArray[ptcounter] = analyselocationArray[resolveArray[rlt]];
                ptcounter++;}
            //to make sure the search stops after one hit
            mt = 12 + 1;
        }
    }
}
//last number of each line could not be compared before, only with
next if-statement
if(resolveArray[rlt] > timeArray[mt]) {goto force;}
else {cout << 
Error! No transport mode found for current match: " << rlt << 
. Press key to continue..."; system("pause>nul");}
}

if(overview)
{
   cout << 
Results saved in file: " << outname;
   output.close();
}

//statistics output
if (!couterstat)
{
   outname2 = "results/stats.txt";
   stats.open(outname2, ios::app);
   pcar = pcar * 100 / ptotal;
   pbike = pbike * 100 / ptotal;
   ppt = ppt * 100 / ptotal;
   stats << 
Statistics of: " << fname << 
Input File total movements: " << ptotal << 
Car%: " << pcar << " Bike%: " << pbike << " PT%: " << ppt << 
counterstat = true;
stats.close();
}

if(!timeframe) stats << "Output File time: " << time << " total movements: " << ptotal
<< "\n";
if(timeframe) stats << "Output File time: " << time << "-" << time2 << " total
movements: " << ptotal << "\n";
stats << "Car%: " << pcar << " Bike%: " << pbike << " PT%: " << ppt << 
stats.close();

//simplify results for cars
if(!timeframe){stats << "results/car" << time << " " << fname;}
if(timeframe){stats << "results/car" << time2 << " " << fname;}
outname = cc.str();
car.open(outname);
outname2 = "results/car.txt";
car2.open(outname2);
sort(carArray, carArray + carcounter);
for(int xx = 0; xx < carcounter; ++xx)
{
   if(carArray[xx] == carArray[xx+1]) {carz+;}
   else {car << carArray[xx] << "," << carz*weight << 
   car2 << carArray[xx] << "," << carz*weight << 
   carz = 1;}
}
cout << 
Results saved in file: " << outname;
car.close();
car2.close();

//simplify results for bikes
if(!timeframe){bb << "results/bike" << time << " " << fname;}
if(timeframe){bb << "results/bike" << time2 << " " << fname;}

outname = bb.str();
bike.open(outname);
outname2 = "results/bike.txt";
bike2.open(outname2);

sort(bikeArray, bikeArray + bikecounter);

for(int xx = 0; xx < bikecounter; ++xx)
{
    if(bikeArray[xx] == bikeArray[xx+1]) {bikez++;}
    else {bike << bikeArray[xx] << "," << bikez*weight << "\n"; bike2 << bikeArray[xx] << "," << bikez*weight << "\n"; bikez = 1;}
}

cout << \nResults saved in file: " << outname;
bike.close();
bike2.close();

//simplify results for pt
if(!timeframe){pt << "results/pt" << time << "," << fname;}
if(timeframe){pt << "results/pt" << time << "," << time2 << "," << fname;}
outname = pt.str();
pubt.open(outname);
outname2 = "results/pt.txt";
pubt2.open(outname2);

sort(ptArray, ptArray + ptcounter);

for(int xx = 0; xx < ptcounter; ++xx)
{
    if(ptArray[xx] == ptArray[xx+1]) {ptz++;}
    else {pubt << ptArray[xx] << "," << ptz*weight << "\n"; pubt2 << ptArray[xx] << "," << ptz*weight << "\n"; ptz = 1;}
}

cout << \nResults saved in file: " << outname << "\n";
pubt.close();
pubt2.close();

cout << \n\nStatistics for current time(frame):
Carcounter: " << carcounter << " Bikecounter: " << bikecounter << " PTcounter: " << ptcounter << " Total: " << ptotal << "\n";

//again?
again:

cout << \nDo you want to enter a new time [y/n]?\n";
cin >> yn;
if (yn == "y" || yn == "n")
{
    if (yn == "y")
    {
        cout << "\n",
        l0000p1:
        cin.clear();
        cin.ignore(INT_MAX, '\n');
        cout << "Enter research time by [hours] [space/enter] [minutes]:\n";
        cin >> hours;
        if(hours < 0 || hours > 23) {goto invaaaa;}
        cin >> minutes;
        if(minutes < 0 || minutes > 59) {goto invaaaa;}
        time = hours * 60 + minutes;
        if (time > 10 && time < 1440)
        {
            cout << Valid input. Time in minutes: " << time << "\n";
            num = 0;
            err = 0;
            carz = 1;
        }
    }
}
bikez = 1;
ptz = 1;
carcounter = 0;
bikecounter = 0;
ptcounter = 0;

if (timeframe)
{
    invaaa2:
    cin.clear();
cin.ignore(INT_MAX, '\n');
cout << "Enter second research time by [hours] [space/enter] [minutes]">;
cin >> hours2;
if(hours2 < 0 || hours2 > 23) {goto mss;}
    
    cin >> minutes2;
if(minutes2 < 0 || minutes2 > 59) {goto mss;}
    time2 = hours2 * 60 + minutes2;
if (time2 <= 10 || time2 >= 1440 || time >= time2)
{
    mss:
    cout << "Invalid input! You have to enter a time between [" << hours << ":" << minutes << "] and [24:00].\n\n";
go to invaaa2;
    }

time2 << "\n\n";
go to backin;
}
else
{
    invaaa:
    cout << "Invalid input! You have to enter a time between [0:10] and [24:00].\n\n"; goto loooop1;
}
}
if (yn == "n")
{
go to end;
}
else
{
cout << "Can not understand the given answer. Please enter [y] or [n].\n";
go to again;
}

//end of program
end:
end = clock() - start;
printf ("\n\nTime built model: %.0f minutes.\n",((float)end)/CLOCKS_PER_SEC)/60);

if(!excel){cout << "\nEnd of program... Press key to exit";}
if(excel){cout << "\nEnd of program... Press key to execute Excel";}
system("pause\nul");

//execute Excel, to handle TransCAD and print created data to pdf-map
if(excel){system("d:/Desktop/TransCad_from_Excel/TransCad_from_Excel/CallingTransCad.xls m");}
end2:
return 0;
10. GISDK Script to Visualize the Analyzed Output Using TransCAD

Macro "merge_to_pdf" (folder_maps, map_name)

    vmapabas=folder_maps+map_name  
    OpenMap(vmapabas,)

    tmp =  
    OpenTable("TempView","CSV",("D:\Documents\Master\Afstuderen\Test\Datatransfer2\Datatransfer2\results\bike.txt"))  
    jointmp = JoinViews("JoinView","Node.node_id","TempView.FIELD_1",{"A"},)  
    SetView("JoinView")  
    SetRecordsValues(null,{{"[MSN]"},null},"Formula",{{FIELD_2},})  
    CloseView("TempView")  
    CloseView("JoinView")  
    RedrawMap(null)

    tmp2 =  
    OpenTable("TempView2","CSV",("D:\Documents\Master\Afstuderen\Test\Datatransfer2\Datatransfer2\results\pt.txt"))  
    jointmp2 = JoinViews("JoinView2","PT.DATA","TempView2.FIELD_1",{"A"},)  
    SetView("JoinView2")  
    SetRecordsValues(null,{{"[MSN]"},null},"Formula",{{FIELD_2},})  
    CloseView("TempView2")  
    CloseView("JoinView2")  
    RedrawMap(null)

    view = "Node"  
    view_set = "Node|"  
    fields_arrays = GetEditableFields(view, "Numeric",)  
    num_fields = ArrayLength(fields_arrays[1])  
    rec = GetFirstRecord(view_set, )  
    while rec <> null do  
        values_array = GetRecordValues(view, rec, fields_arrays[1])  
        for i = 1 to num_fields do  
            if values_array[i][2] = null then values_array[i][2] = 0  
        end  
        SetRecordValues(view, rec, values_array)  
        rec = GetNextRecord(view_set, rec, )  
    end

    view = "PT"  
    view_set = "PT|"  
    fields_arrays = GetEditableFields(view, "Numeric",)  
    num_fields = ArrayLength(fields_arrays[1])  
    rec = GetFirstRecord(view_set, )  
    while rec <> null do  
        values_array = GetRecordValues(view, rec, fields_arrays[1])  
        for i = 1 to num_fields do  
            if values_array[i][2] = null then values_array[i][2] = 0  
        end  
        SetRecordValues(view, rec, values_array)  
        rec = GetNextRecord(view_set, rec, )  
    end  
    RedrawMap(null)

    a = GetLegendSettings()
11. Excel Visual Basic script to execute the GISDK script

Sub Saves_Original()

' Macro Parameters
project_folder = "D:\Documents\Master\Afstuderen\7MM37\transcad and new scenario\Original\"
macro_folder = "D:\Desktop\TransCad_from_Excel\TransCad_from_Excel\Macro Folder\"
map = "GrApp.map"
macro_name = "merge_to_pdf"

' Opens TransCad and sets the automation server
Shell "C:\Program Files\TransCAD\tcw.exe"
Set TCW = CreateObject("TransCAD.AutomationServer")
retval = TCW.Macro(macro_name, macro_folder + "macro.rsc", project_folder, map)

'We close the notepads and tcw that are open now after the tcw's macro ran
Application.SendKeys "%{F4}", True
Application.SendKeys "%{F4}", True
Application.SendKeys "%{F4}", True

End Sub

The Excel Visual Basic script for execution of the scenario is similar to the coding above, except 'Original' has to be replace for 'Scenario'.

12. MATLAB SCRIPT TO CHECK AND CHANGE ACTIVITY PROGRAM DATA (MAAS & LIAO, 2015)

k=0;

%part 1
ind = find(a(:,1)==0);
ind(end+1) = size(a,1)+1;
emptyArray = zeros(10000,1);
empArrLen = 0;
for i=1:length(ind)-1
    begP = ind(i)+1;
    endP = ind(i+1)-1;
    totalT = sum(a(begP:endP,3));
    if totalT > 800
        k=k+1;
    end
    for j=endP:-1:begP+1
        if a(j,2)==-1 && a(j,1)==a(j-1,1)
            empArrLen = empArrLen+1;
            emptyArray(empArrLen) = j;
            if a(j-1,3) == a(j,3) || totalT > 800
                a(j-1,3) = a(j-1,3);
            else
                a(j-1,3) = a(j-1,3)+a(j,3);
            end
        end
    end
end
emptyArray = emptyArray(1:empArrLen);
emptyArray;
a(emptyArray,:) = [];

%part 2.1
ind = find(a(:,1)==0);
ind(end+1) = size(a,1)+1;
e = diff(ind)-1;
z = find(e > 6)
delIndex = [];
sttP = ind(z);
endP = ind(z+1);
for i=1:length(z)
    delIndex = [delIndex, sttP(i)+6:endP(i)-1];
end
delIndex
a(delIndex,:)=[];

%part 2.2
ind = find(a(:,1)==0);
ind(end+1) = size(a,1)+1;
emptyArray = zeros(10000,1);
empArrLen = 0;

for i=1:length(ind)-1
    begP = ind(i)+1;
    endP = ind(i+1)-1;
    totalT = sum(a(begP:endP,3));
    if totalT > 800
        k=k+1;
    end
    for j=endP:-1:begP+1
        if a(j,2) > 0 && a(j,1) == a(j-1,1) && a(j-1,3) == a(j,3)
            empArrLen = empArrLen+1;
            emptyArray(empArrLen) = j;
            a(j-1,3) = a(j-1,3);
        end
    end
end
emptyArray = emptyArray(1:empArrLen);
emptyArray;
a(emptyArray,:) = [];

%part 3
ind = find(a(:,1)==0);
ind(end+1) = size(a,1)+1;
for i=1:length(ind)-1
    len = ind(i+1)-ind(i)-1;
    a(ind(i)+1,4) = 0;
    a(ind(i)+len,5) = 0;
    for j=len
        a(ind(i)+j,4) = a(ind(i)+j-1,4) + 2^(j-2);
    end
    for j=len-1:-1:1
        a(ind(i)+j,5) = a(ind(i)+j+1,5) + 2^j;
    end
end
b=a;

%part 4
ind = find(a(:,1)==0);
ind(end+1) = size(a,1)+1;
b(ind(1:end-1,3)=diff(ind)-1;

%part 5
d=b(:,1)==0;
b(d,2)=0:sum(d)-1;
13. SPSS SCRIPT TO MERGE ALL SEPARATE PT FILES INTO ONE LARGE DATABASE

GET FILE="stops.sav".
SORT CASES BY stop_id.
SAVE OUTFILE="stop.s".
GET FILE="stop_times.sav".
SORT CASES BY stop_id.
SAVE OUTFILE="stop_times2.sav".
MATCH FILES FILE="stop_times2.sav" /TABLE="stops2.sav" /BY stop_id.
SORT CASES BY trip_id.
SAVE OUTFILE="merge.sav".

GET FILE="routes.sav".
SORT CASES BY route_id.
SAVE OUTFILE="routes2.sav".
GET FILE="trips.sav".
SORT CASES BY route_id.
SAVE OUTFILE="trips2.sav".
MATCH FILES FILE="trips2.sav" /TABLE="routes2.sav" /BY route_id.
SORT CASES BY route_id.
SAVE OUTFILE="merge2.sav".

GET FILE="merge2.sav".
SORT CASES BY trip_id.
SAVE OUTFILE="merge2b.sav".
GET FILE="merge.sav".
SORT CASES BY trip_id.
SAVE OUTFILE="mergeb.sav".
MATCH FILES FILE="mergeb.sav" /TABLE="merge2b.sav" /BY trip_id.
SORT CASES BY route_id.
SAVE OUTFILE="merge3.sav".

14. SETTINGS OF STANDARD ADOBE PDF PRINTER

![Adobe PDF Printing Preferences](image)
15. Declarations specified in ‘declare.h’ called by the visualization application

//CUSTOMIZABLES
bool fastmode = true; // not asked in model
    // used for registration whether or not debug-mode is chosen (true = fast, false = debug)
bool timeframe = true; // not asked in model
    // used for registration whether or not a timeframe is used (true = on, false = off)
bool overview = false; // not asked in model
    // used for registration whether or not create an overview file (true = yes, false = no)
b外面 excel = false; // not asked in model
    // used for registration whether or not Excel should be executed automatically right
    after the script (true = yes, false = no)
b外面 couter = false; // automatically applied
    // used for registration whether or not a system message appeared (true = disable system
    message)
bool couterstat = false; // automatically applied
    // used for registration whether or not the statistics of the input file are printed to a
    file already (true = disable print)
int oddeven = 0; // not asked in model
    // used for specification of the input file structure (0 = file starts with location
    value, 1 = file starts with time value)
int weight = 15; // not asked in model
    // used for specification of the weight of each person within the project (1 = no weight)

//ADVANCED VALUES
int *countArray = new int[300000]; // saved amount of words on each lines of
    // input file - Should be higher or equal to number of lines of the inputfile !
int *locationArray = new int[150000]; // saved number of value within file which
    // is a starter for location line - Should be half of size as countArray !
int *timeArray = new int[150000]; // saved number of value within file which
    // is a starter for time line - Should be half of size as countArray !
int *resolveArray = new int[1000000]; // saved number of value within file where
    // input time is valid - Should be higher or equal to assumed volume of total solutions !
string *analysetimeArray = new string[7500000]; // saved time values - Should be higher or
    // equal to number of time values of the inputfile or half of total words of the inputfile !
string *analyselocationArray = new string[7500000]; // saved location values - Should be higher or
    // equal to number of location values of the inputfile or half of total words of the inputfile !
double *analtimeArray = new double[7500000]; // copy of the string analysetimeArray to
double - Should be equal to analysetimeArray !

string *carArray = new string[500000]; // saved results for car locations on the
given time - Should be higher or equal to assumed volume of car solutions !
string *bikeArray = new string[500000]; // saved results for bike locations on the
given time - Should be higher or equal to assumed volume of bike solutions !
string *ptArray = new string[500000]; // saved results for pt locations on the
given time - Should be higher or equal to assumed volume of PT solutions !

//FIXED VALUES
string fname; // input from user, filename input data
string yn; // input from user, whether or not continue script with another time
ifstream file; // used for the input file
int time; // calculated from input user, first time to be researched in minutes
int time2; // calculated from input user, second time to be researched in minutes
int hours; // input from user, first time hours
int hours2; // input from user, second time hours
int minutes; // input from user, first time minutes
int minutes2; // input from user, second time minutes
string line; // line content to analyze amount of words on line
int r = 0; // used for current and total number of lines
int ci = -1; // used for current and total number of location values
int cj = -1; // used for current and total number of time values
int count; // used for amount of words on line
int maxcount = 0; // used for the maximum amount of words on line
int wordcounter; // used for amount of words in file
int l1 = -1; // used for current and total number of starters in the location line
int l2 = -1; // used for current and total number of starters in the time line

int num = 0; // used for number of matches match on given time
int err = 0; // used for amount of dropped matches on given time
int carcounter = 0; // used for current and total number of car solutions
int bikecounter = 0; // used for current and total number of bike solutions
int ptcounter = 0; // used for current and total number of pt solutions
int carz = 1; // used for the amount of individuals on the same location by car
int bikez = 1; // used for the amount of individuals on the same location by bike
int ptz = 1; // used for the amount of individuals on the same location by pt
ostringstream oo; // used for the output overview file
ostringstream cc; // used for the output car file
ostringstream bb; // used for the output bike file
ostringstream pt; // used for the output pt file
string outname; // used for the output overview, car, bike and pt files
string outname2; // used for the output stats files
ofstream output; // used for the output overview file
ofstream car; // used for the output car file
ofstream car2; // used for the output car file
ofstream bike; // used for the output bike file
ofstream bike2; // used for the output bike file
ofstream pubt; // used for the output pt file
ofstream pubt2; // used for the output pt file
ofstream stats; // used for the output stats file
int ptotal = 0; // used for percentage of traffic for statistics
int pcar = 0; // used for percentage of car traffic for statistics
int pbike = 0; // used for percentage of bike traffic for statistics
int ppt = 0; // used for percentage of pt traffic for statistics
int P; // used for current transport mode to create statistics