

TUM

FAKULTÄT FÜR BAUINGENIEUR- UND VERMESSUNGSWESEN
TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor Thesis
Lehrstuhl für Computation in Engineering

Evaluierung des Verkehrssimulators MATSim

Bearbeiterin: Alesya Zhitova

Lehrstuhlinhaber: Prof. Dr. rer. nat. Ernst Rank

Betreuer TUM: Dr.-Ing. André Borrmann, Dipl.-Inf. Angelika Kneidl

Betreuer Siemens: Dr. Wolfram Klein

Abgabetermin: 28.07.2009

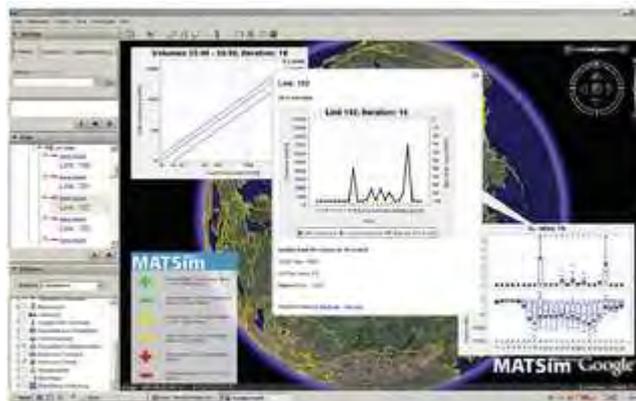
Aufgabenstellung

Technische Universität München
Chair for Computation in Engineering

Prof. Dr. rer. nat. Ernst Rank

Bachelor-Arbeit **Evaluierung des Verkehrssimulators MATSim**

Gegenstand der Bachelorarbeit ist die Untersuchung des als OpenSource-Software verfügbaren Verkehrssimulators MATSim auf Eignung zur Kopplung mit dem im Forschungsvorhaben REPKA zu entwickelnden Personenstromsimulator. Dabei sollen zum einen die mathematischen Grundlagen des Simulators und zum anderen der Aufbau der Software und ihre Leistungsfähigkeit analysiert werden.



Die Bearbeitung gliedert sich in folgende Teilaufgaben:

1. Dokumentation der Installationsprozedur
2. Einarbeiten und Test der Software (Test einfacher Topographien, Generierung von Verkehr)
3. Analyse und Dokumentation des mathematisches Modells anhand der vorher erzielten Ergebnisse und der verfügbaren Dokumentation
4. Test auf Grenzen des Modells (Welche Verkehrselemente können nicht modelliert werden (Ampeln, Kreuzungen))?
5. Laufzeit- und Speicherplatzuntersuchungen
6. Formulieren von Aussagen zur Realitätsnähe
7. Theoretisches Erarbeiten gemeinsamer Szenarien für MATSim und SinoSim

Bearbeitung: Alesya Zhitova

Betreuung TUM: Dr.-Ing. André Borrmann, Dipl.-Inf. Angelika Kneidl

Betreuung Siemens: Dr. Wolfram Klein

Zusammenfassung

MATSim-T (**M**ulti-**A**gent **T**ransport **S**imulation **T**oolkit) ist ein als OpenSource-Software verfügbarer Verkehrssimulator. Er führt, basierend auf einem agentengestützten Modell, umfangreiche Simulationen durch und bildet dadurch das dynamische Verkehrsaufkommen realitätsnah ab.

Außerdem gehört zu seinen Vorteilen unter anderem Folgendes:

- der modulare Aufbau und die Möglichkeit, durch eigene projektbezogene Module und Implementierungen seine Leistungsfähigkeit zu steigern;
- die Möglichkeit, Zwischenresultate der Simulation zu jedem Zeitpunkt einzusehen;
- der zur freien Verfügung gestellte Code in der Java-Sprache.

Nachteilig ist dagegen, dass:

- der Simulator sich noch in der Entwicklungsphase befindet und nicht alle Simulationsbedürfnisse abdecken kann und
- die Rechenzeit, der Speicherplatzbedarf und der Aufwand der Implementierung von Objekten mit der wachsenden Zahl der bis zum Erreichen des stabilen Zustandes benötigten Optimierungsiterationen erheblich ansteigen.

Aufgrund der grundlegenden Unterschiede zwischen MATSim und dem im Rahmen des Forschungsvorhabens REPKA zu entwickelnden Personenstromsimulator SinoSim hinsichtlich der Zielsetzung und des konzeptionellen Aufbaus wäre theoretisch fast ausschließlich nur der Abgleich der Simulationsergebnisse und gegebenenfalls eine Wechselwirkung im Bereich der Optimierung denkbar.

Die Arbeit stellt eine Grundlage für weitere detailliertere Untersuchungen des Verkehrssimulators MATSim dar, sofern die REPKA-Projektentwickler dafür weiteren Bedarf sehen.

Abstract

MATSim-T (**M**ulti-**A**gent **T**ransport **S**imulation **T**oolkit) is a transport simulator available as open-source software. The software consists of several modules which can be used together or as stand-alone tools. The current functionality allows for agent-based traffic flow simulations on large-scale scenarios and the ability to conduct analyses with real-world traffic counting stations.

The key features of MATSim are:

- A modular approach provides the ability to increase the efficiency of the simulator by utilizing project specific modules and implementations.
- The ability to visualize detailed intermediate data produced during the simulation.
- The availability of the complete Java source code at no cost.

The disadvantages of MATSim are:

- The simulator is still in the development stage and cannot currently cover all simulation needs.
- Potential for large computing times and memory requirements may prevent users with limited computing power from completing the required analyses.
- Implementation of new objects becomes increasingly time consuming and complex as the number of optimization iterations increase to reach stable state.

Although there are fundamental differences between MATSim and the pedestrian stream simulator SinoSim, developed in the context of the REPKA research project, concerning their purposes and conceptual structure, it would be theoretically conceivable to compare the results from both simulations to achieve potentially increased optimization.

This paper offers basic information to form the base of more detailed investigations of the traffic simulator MATSim and its use to enhance the REPKA-project.

Inhaltsverzeichnis

1 Einführung in die Arbeitsweise des Verkehrssimulators MATSim.....	6
1.1 Allgemeines über den Verkehrssimulator MATSim.....	6
1.2 Konzeptioneller Aufbau des Simulators.....	11
1.2.1 Generieren einer Simulation.....	13
1.2.2 Events.....	16
1.2.3 Module.....	18
1. 3 Grenzen und Realitätsnähe des Modells.....	19
2 Laufzeit- und Speicherplatzuntersuchungen.....	26
3 Technische Details zum Verkehrssimulator MATSim.....	34
3.1 Installation des Simulators.....	34
3.2 Einarbeiten und Test des Simulators anhand eines Beispiels.....	35
4 Kopplungsmöglichkeiten von MATSim und SinoSim.....	40
5 Zusammenfassung und Ausblick.....	43

Literaturverzeichnis

Erklärung

Anhang A: Analyse der für die Optimierung benötigten Anzahl der Iterationen

Anhang B: Publikationsverzeichnis

Anhang C: Abbildungsverzeichnis

Anhang D: Tabellenverzeichnis

Anhang E: Inhaltsverzeichnis der beigefügten CD

1 Einführung in die Arbeitsweise des Verkehrssimulators MATSim

1.1 Allgemeines über den Verkehrssimulator MATSim

MATSim-T (**M**ulti-**A**gent **T**ransport **S**imulation **T**oolkit) ist ein als OpenSource-Software verfügbarer Verkehrssimulator, der vor allem von folgenden zwei Personengruppen entwickelt wurde:

- Verkehrssystemplanung und Verkehrstelematik am Institut für Land- und Seeverkehr (ILS), Technische Universität Berlin, unter der Leitung von Prof. Dr. Kai Nagel und
- Verkehrsplanung am Institut für Verkehrsplanung und Transportsysteme (IVT), Eidgenössische Technische Hochschule Zürich, unter der Leitung von Prof. Dr. Kay W. Axhausen.

Durch die Eingabe von <http://www.matsim.org> gelangt man auf die Homepage des Simulators:



Abbildung 1: Homepage des Verkehrssimulators MATSim (<http://www.matsim.org>).

Der Abbildung ist zu entnehmen, dass die Seite sehr übersichtlich aufgebaut ist. Neben der allgemeinen Information zum Verkehrssimulator und der Beschreibung seiner Features findet man auf den ersten Blick auch:

- die umfangreiche Dokumentation, die unter anderem auch die Anleitungen zum Handhaben des Programms für seine Benutzer und die Weiterentwickler, sowohl Tutorien zum Einstieg und den ersten Schritten bei der Arbeit mit dem Simulator enthält:

MATSim
Multi-agent Transport Simulation Toolkit

Search

- Home
- Documentation**
 - User's Guide
 - Developer's Guide
 - Tutorials
 - FAQ
- Downloads
- Examples
- Publications
- About us

Links

Technische Universität Berlin

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Laboratoire d'Economie des Transports

SOURCEFORGE.NET

DOCUMENTATION

If you are new to MATSim, we suggest that have a look at the [tutorial](#) "Getting started with MATSim". It will teach you the basics of MATSim and give you an introduction to the code. Once you start writing your own classes that either make use of MATSim or otherwise interact with MATSim, you may be interested in our notes in the [Developer's Guide](#).

USER'S GUIDE

In the User Guide, you will find information about how to use MATSim, how to enable special features, and more.

- > [Building a new Scenario](#)
- > [Features](#)
- > [System Requirements](#)
- > [Time dependent networks](#)
- > [Using OTFVis](#)

TUTORIALS

We have prepared some tutorials which should help you getting started with MATSim.

- > [Getting Started with MATSim](#)
- > [Using DEQSim with MATSim](#)
- > [Using Nightly Builds of MATSim](#)
- > [Evacuation Tutorial](#)

FAQ

Have a look at [frequently asked questions](#), and their answers.

DEVELOPER'S GUIDE

The developer guide

- > provides specification of key-aspects of MATSim (e.g. file formats, simulation events, ...)
- > lists some guidelines for developers (e.g. coding conventions) to keep MATSim maintainable.
- > gives information about certain packages, how to (programmatically) use them and what features they offer.
- > offers additional pieces of information that may/should be of interest to developers (e.g. how to use Eclipse for development, or some Java-tips)

Choose a topic of interest to read more:

- > [Getting and Building the Code](#)
- > [Concepts](#)
- > [Contributing / Writing Code](#)
- > [File Formats](#)
- > [Modules / Features](#)

Abbildung 2: Inhaltsverzeichnis der Dokumentation zum Verkehrssimulator MATSim.

- Downloads,
- Beispiele,
- Publikationen (Auflistung der Publikationen ist im Anhang B zu finden),
- Impressum und

- Neuigkeiten: Da der Simulator MATSim ständig weiterentwickelt wird (z.B. es wird momentan unter anderem an neuen Events¹ gearbeitet), sind auf der Homepage auch die Neuerungen und die Überarbeitungen und Ergänzungen zu finden.

Eine kompakte, allgemeine Beschreibung der Leistungsfähigkeit des Simulators findet sich in der Diplomarbeit von Neumann [1] wieder: „MATSim ist in der Lage, Szenarios² mit bis zu 8 Millionen Agenten gleichzeitig zu simulieren und dabei Netze zu verwenden, die mehrere hunderttausend Kanten umfassen und beispielsweise die gesamte Schweiz abdecken. Dabei hat jeder einzelne Agent einen bis mehrere Pläne, welche verschiedene Aktivitäten enthalten. Diese Pläne werden für alle Agenten während der Verkehrssimulation simultan ausgeführt. Werden mehrere Simulationen hintereinander ausgeführt, so sind die Agenten in der Lage, zwischen den Simulationen ihre Pläne zu modifizieren.“ Außerdem ist zu erwähnen, dass mit MATSim solche Simulationen für einige Tage im Minuten- oder Sekundentakt geschrieben werden können. Als Beispiele werden dabei auf der Homepage folgende Anwendungen des Simulators angeführt:

- Verkehrssimulation der gesamten Verkehrsnachfrage der Schweiz. Modelliert wurde ein Tag der gesamten schweizerischen Population (nähere Beschreibung dieses Beispiels s. auch unter Kapitel 2 „Laufzeit- und Speicherplatzuntersuchungen“). Als Beispiel wurden zehn Prozent der Nachfrage (etwa 180.000 Agenten) in Form einer pdf-Datei (Zoomen ins Zentrum der Metropole Zürich möglich) visualisiert. Jeder bunte Punkt repräsentiert dabei ein einzelnes simuliertes Verkehrsmittel (in dem Fall einen Pkw), die grüne Farbe bedeutet, dass es sich mit hoher Geschwindigkeit fortbewegt, rote – es kommt zum stockenden Verkehr:

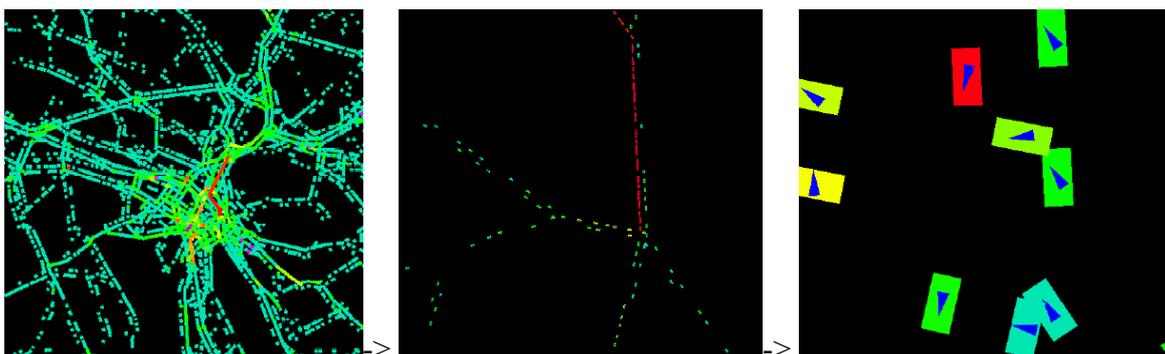


Abbildung 3: Beispiel der Anwendung des Verkehrssimulators MATSim zur Simulation der gesamten schweizerischen Verkehrsnachfrage. Ausschnitte aus der pdf-Visualisierungsdatei. Quelle: <http://matsim.org/uploads/showcase/MATVis-Zoom.pdf>

¹ Vgl. Kapitel 1.2.2.

² Generieren der Szenarien wird im Kapitel 1.2.1 ausführlich behandelt.

- Das nächste Beispiel zeigt die Durchquerung des Baregg-Tunnels in der Schweiz im 30-Minuten-Takt³. Die Visualisierung wurde mit Hilfe eines Geographischen Informationssystems (GIS) durchgeführt:

Dynamische Spinne: Baregg Tunnel, Richtung Zürich, 07:00 – 08:00 Uhr

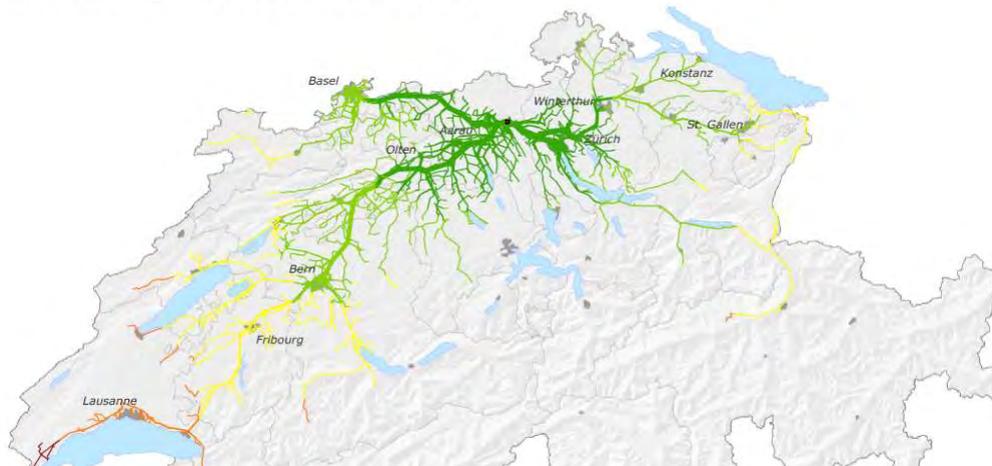


Abbildung 4: Beispiel der Anwendung des Verkehrssimulators MATSim zur Simulation der Durchquerung des Baregg-Tunnels in der Schweiz.

Quelle: http://matsim.org/uploads/showcase/Spinne_Baregg0700.pdf

- Das letzte zu nennende Beispiel der mit MATSim bereits realisierten Projekte zeigt die Simulation der Verkehrsnachfrage in Berlin und seiner Umgebung (150 x 250 km). Das modellierte Netz besteht aus 10.000 Knoten und 30.000 Kanten/Verbindungen. Visualisiert wurden wiederum zehn Prozent der simulierten Nachfrage, d.h. etwa 160.000 Agenten:

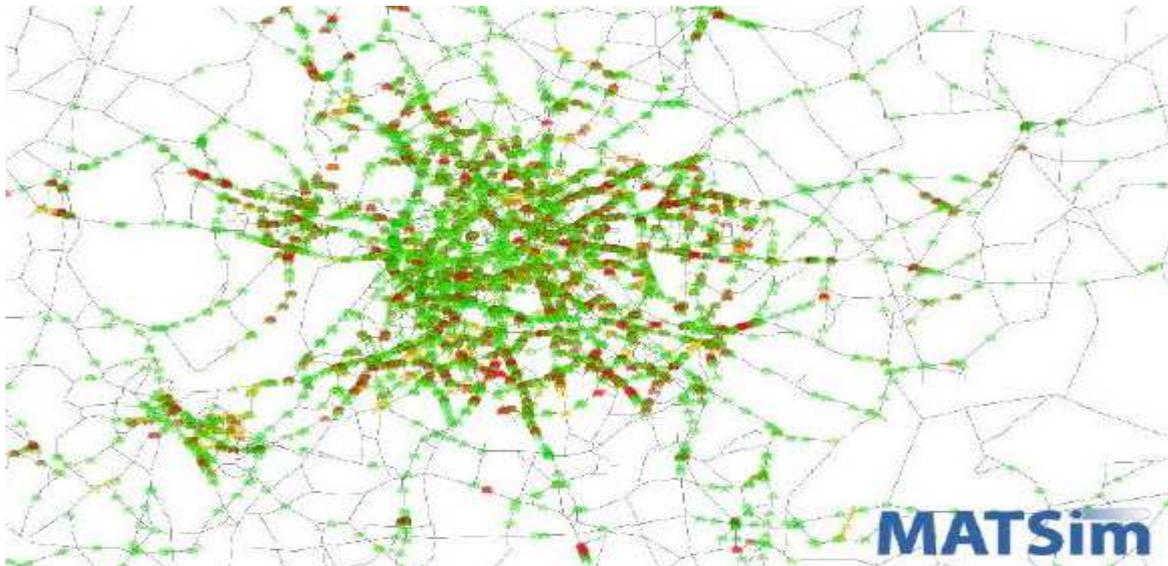


Abbildung 5: Beispiel der Anwendung des Verkehrssimulators MATSim zur Simulation der gesamten Verkehrsnachfrage Berlins und seiner Umgebung.

Quelle: <http://www.matsim.org/scenario/berlin>

³ "The example shows the paths of all travellers crossing the "Baregg tunnel" heading towards Zurich, Switzerland, between 7am and 8am ("Dynamic Spider-Analysis" / "Dynamische Spinnen-Analyse"). The colours show where the agents were earlier (on the left side of the tunnel) or later (on the right side) in 30min steps."

Quelle: <http://www.matsim.org/examples>

Den aufgeführten Beispielen, sowohl der vorhandenen Dokumentation kann entnommen werden, dass die MATSim-Entwickler sich zum Ziel gesetzt haben, mit dem neuen Verkehrssimulator MATSim, dem 4-Stufen-Algorithmus, das bekanntlich die Verkehrsplanung in vier Schritten (Verkehrserzeugung, Verkehrsverteilung, Verkehrsmittelwahl und Routenwahl) beschreibt, eine Alternative zu bieten. Dieser Algorithmus stößt an seine Grenzen, da er den Verkehr nicht zeitabhängig erfassen kann und nicht die subjektiven menschlichen Entscheidungen berücksichtigt (vgl. Rieser [2]). Balmer et al. [3] weisen in diesem Zusammenhang darauf hin, dass *„die Ansprüche der Verkehrsplanung jedoch nicht das detaillierte Fahrverhalten sind, sondern eine realitätsnahe Abbildung des dynamischen Verkehrsaufkommens, der Staubildung und des daraus entstehenden Rückstaus, die Stauauflösung, usw. Diese Anforderungen erfüllt das Warteschlangen Modell. Dabei wird jede Straße durch eine Warteschlange modelliert, in der Fahrzeuge mindestens die benötigte Fahrzeit auf freier Straße abwarten müssen. MATSim-T stellt zwei Implementierungen dieses Modells zur Verfügung, die für die realistische Umsetzung der Nachfrage notwendigen Anforderungen erfüllen. Sie beachten die Randbedingungen des Verkehrs (Kapazität einer Straße, Höchstgeschwindigkeit, etc.), der infrastrukturellen Randbedingungen (z.B. Öffnungszeiten einer Aktivität), wie auch den physikalisch machbaren Rahmen. Als Resultat produziert die Simulation für jeden Agent, zu jedem Zeitpunkt des gesamten Tages Informationen darüber, wo dieser gerade ist, und was er gerade macht.“* Aus der Summe aller Agenten kristallisiert sich wiederum ein Verkehrszustand/Verkehrsaufkommen heraus, das als Alternative zu dem bereits oben erwähnten 4-Stufen-Algorithmus gesehen werden kann. Den Unterschied zwischen den traditionellen Ansätzen des 4-Stufen-Algorithmus und der Herangehensweise des Verkehrssimulators MATSim betonen auch Meister et al. [4]: *„Verschiedene Methoden verfolgen den gleichen Modellzweck einer tageszeitlich variierenden Verkehrsnachfrage wie MATSim. Sie basieren jedoch nicht auf dem agentenbasierten Ansatz mit zeitlich konsistenten Aktivitätenplänen, sondern auf in Quelle-Ziel-Matrizen aggregierten isolierten Wegen wie sie in klassischen statischen Verkehrsmodellen eingesetzt werden. Diese Matrizen werden mit verschiedenen Methoden über den modellierten Durchschnittstag verteilt und auf das Modell des Verkehrsnetzes angewandt.“* Das primäre Ziel des Verkehrssimulators MATSim ist somit das individualisierte Modellieren jedes einzelnen Agenten und seiner Tagesabläufe und die Simulation des daraus entstehenden Verkehrs.

1.2 Konzeptioneller Aufbau des Simulators

MATSim besteht aus mehreren logischen in sich abgeschlossenen Modulen (s. Abbildung 6). Die Basis bildet dabei das Modul MATSIM-DB, das alle Daten enthält, die für die Beschreibung der Agenten und ihrer Nachfrage zur Verfügung stehen. MATSIM-DATA (mit World-DB, Network-DB, Facilities-DB etc.) dient der Beschreibung der Szenarien. MATSIM-FUSION speichert die Daten der Szenarien auf eine geordnete, zusammenhängende Weise. MATSIM-INI modelliert ein anfängliches, individuelles Szenario für jede einzelne Person, während MATSIM-EA dieses individuelle Szenario jedes einzelnen Agenten in Referenz zu den Szenarien anderer Agenten in der gleichen Simulation, durch Iterationen optimiert.

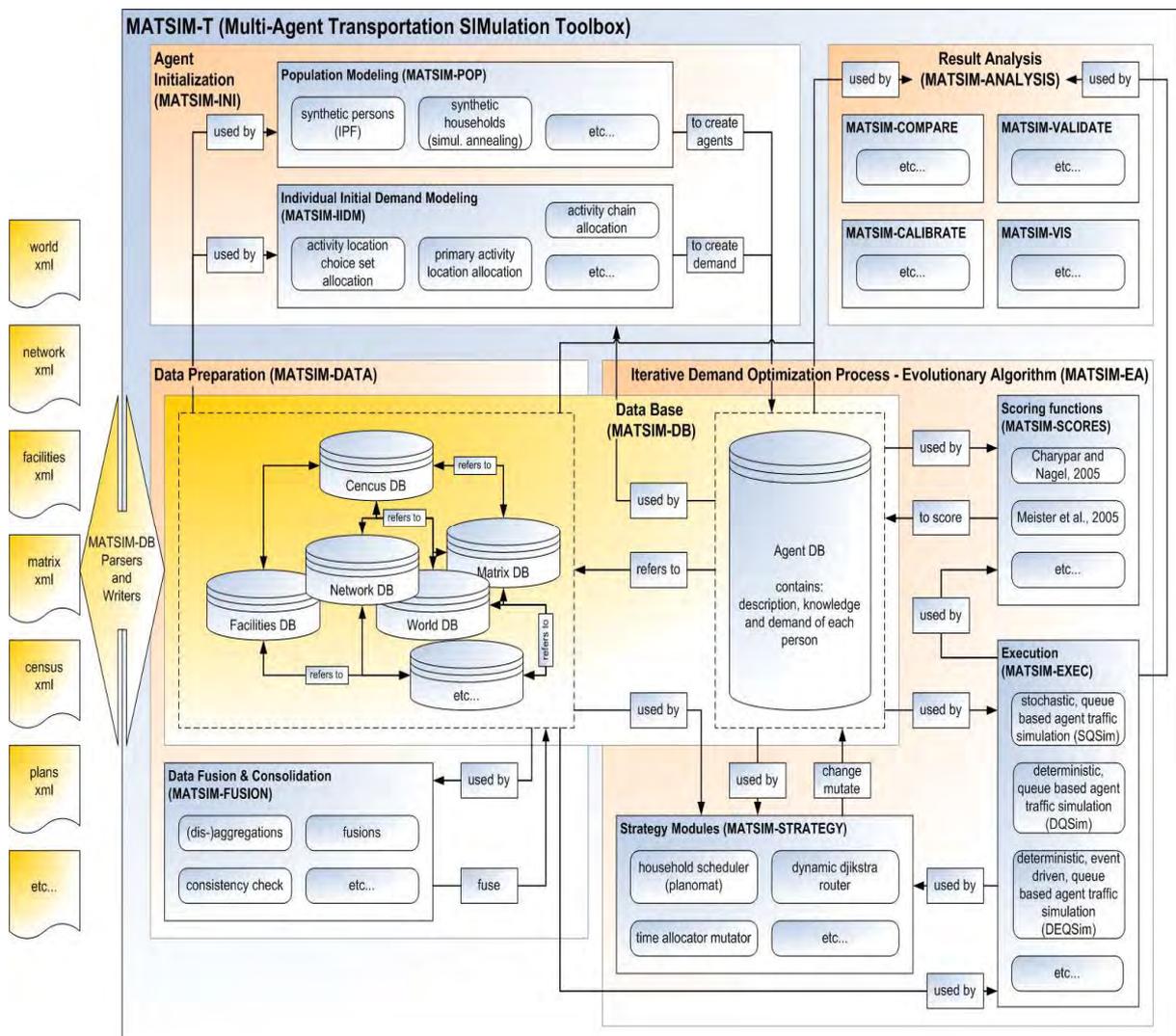


Abbildung 6: MATSim als modulares System.

Quelle: <http://www.matsim.org/uploads/matsim-t.png>

Dabei läuft der Prozess der Simulation in folgenden drei Schritten ab:

1. Modellierung der Agenten und ihrer anfänglichen Nachfragen (Vorbereitung)
2. Optimierung der Nachfragen durch Iterationen (Ausführung) und
3. Analyse der bei jeder Iteration erzeugten Daten (Nachbereitung)

Dabei ist zu bemerken, dass die Zwischenresultate des Modellierungsprozesses zu jedem beliebigen Zeitpunkt einsehbar sind (vgl. Balmer et al. [3]: *„Zu jedem beliebigen Zeitpunkt während des Modellierungsprozesses können zudem die Zwischenresultate ausgegeben werden. Dies ist umso wichtiger, da es typischerweise notwendig ist, die Resultate eines als Modul implementierten Modells statistisch zu validieren. Die Zwischenresultate können wiederum als Ausgangslage für weitere Modellierungsschritte benutzt werden. Ebenso ist es möglich mehrere Prozessschritte hintereinander, „in einem Zug“ auszuführen.“*).

Eine kompakte allgemeine Beschreibung des Modellierungs- und Simulationsprozesses findet sich in der Einführung der Diplomarbeit von A. Neumann [1] wieder: *„Der erste Abschnitt ist die Vorbereitung aller für die Simulation benötigter Eingangsdaten. Als Eingangsdaten dienen eine Netzwerkbeschreibung, Beschreibungen von Einrichtungen wie Arbeitsplätzen und Sportstätten, die zu simulierende Bevölkerung und die daraus resultierende anfängliche Nachfrage. Aus den Eingangsdaten werden für jeden aus der Bevölkerung generierten Agenten Pläne erstellt. Im zweiten Abschnitt, der eigentlichen Simulation, kommen diese Pläne zur Ausführung. Dabei wird genau ein Plan jedes Agenten mit Hilfe der Verkehrssimulation ausgeführt. Im Anschluss wird der ausgeführte Plan eines Agenten bewertet. Da die Ausführung für alle Agenten simultan erfolgt, beeinflussen sich Agenten gegenseitig und konkurrieren um begrenzte Ressourcen wie die Kapazität einer Kante. Nach der Bewertung erhält ein Teil der Agenten die Möglichkeit seinen Plan zu modifizieren. Dieser Vorgang wird als Replanning bezeichnet (lediglich ein Teil der Agenten darf im Anschluss an die Ausführung seinen Plan modifizieren). Zu den Möglichkeiten der Modifikation gehört die Wahl einer anderen Abfahrzeit, um besonders staugefährdete Zeiträume zu meiden. Alternativ kann auch eine neue Route zwischen zwei Aktivitäten gesucht werden. Der wohl am häufigsten benutzte Algorithmus dafür ist der von Dijkstra (1959) entwickelte und nach ihm benannte Dijkstra-Algorithmus zum Finden kürzester Wege. Dessen erweiterte zeitbasierte Variante wird in MATSim verwendet. Die zeitbasierte Variante erlaubt es, die Kosten der Kante in Abhängigkeit des Startzeitpunktes zu variieren. Identische Routen liefern zum Beispiel zu unterschiedlichen Zeitpunkten verschiedene Reisezeiten. Im Anschluss kommen die Pläne erneut zur Ausführung. Durch die Wiederholung der drei Schritte Ausführung, Bewertung und Replanning wird das Lernen der Agenten simuliert. Eine solche Wiederholung wird als Iteration bezeichnet. Der dritte und letzte Abschnitt, die Nachbereitung, befasst sich mit der Analyse der bei jeder Iteration erzeugten Daten.“*

So wurde in Abbildung 7 beispielsweise die synthetische Population mit Hilfe eines externen Programms (Frick, 2004) generiert und in die Datenbank MATSIM-DB übergeben, die alle diese Daten durch MATSIM-FUSION strukturiert, so dass sie als anfängliche Nachfrage

jeder einzelnen Person in MATSIM-INI benutzt werden konnten. Für den Optimierungsprozess wurden diese im XML-Format vorliegenden Daten in eine andere Datenbank MATSIM-DB überführt. Nach der Optimierung im Modul MATSIM-EA wurden die Ergebnisse wiederum in die Dateien (XML und TXT) geschrieben. Diese Dateien werden im weiteren Schritt durch das Modul MATSIM-ANALYSIS ausgewertet. Der Grund der Konvertierung liegt darin, dass einige Teile von MATSim in C++ geschrieben sind. Der Simulator selbst ist zum größten Teil in der Java-Sprache geschrieben und nutzt die Dateien mit dem XML-Format.

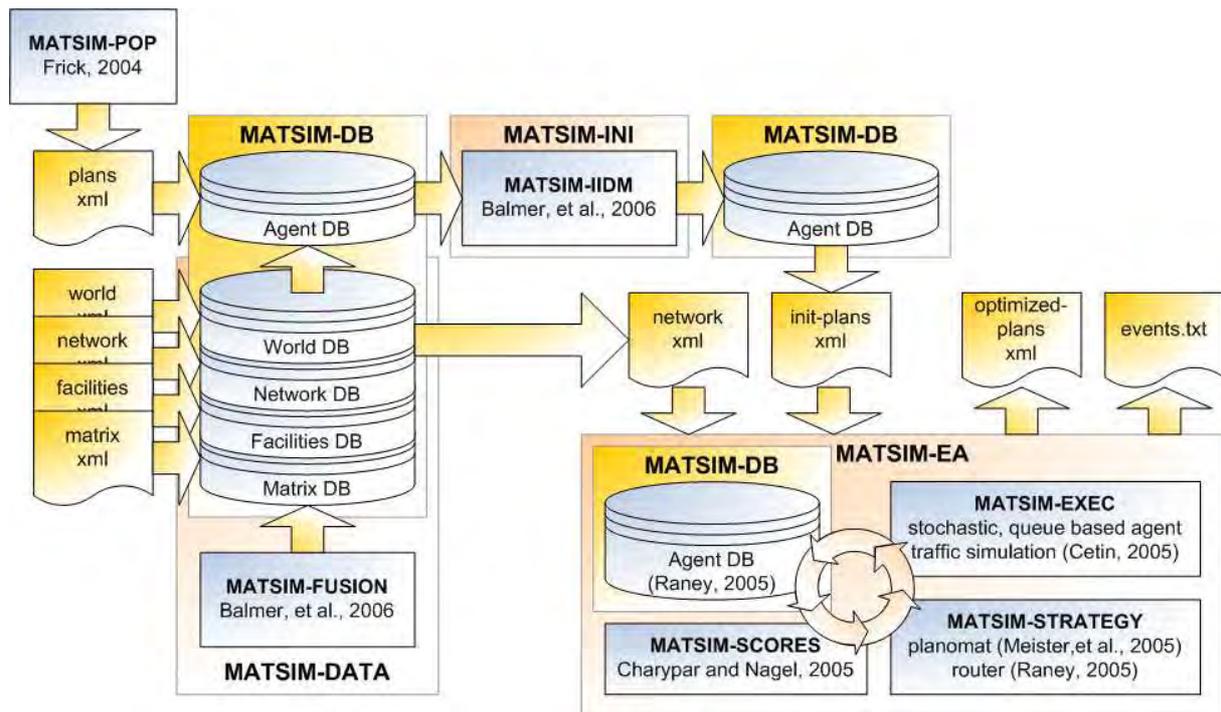


Abbildung 7: Der Prozess einer Simulation auf konzeptioneller Ebene.
Quelle: <http://www.matsim.org/uploads/matsim-t-process.png>

1.2.1 Generieren einer Simulation

Ein neues Szenario zu programmieren kann unter Umständen ein sehr mühsamer und langwieriger Prozess werden, denn es werden viele Daten für eine solide Simulation benötigt. Das Gute an MATSim ist, dass man die Simulation auch mit wenigen einfachen Vorgaben starten kann, die zu einem späteren Zeitpunkt erweitert und weiterentwickelt werden können. In jedem Fall braucht man für ein Szenario zumindest ein Netzwerk und eine Nachfrage (Aktivitäten der synthetischen Population). Die Simulation liefert dann die Informationen über den zeitlichen, dynamischen Tagesablauf jeder einzelnen Person, der individuell und auch in Abstimmung mit allen anderen Agenten optimiert⁴ werden kann. D.h.

⁴ Siehe Kapitel 1.2.1.

zu jedem Zeitpunkt ist bekannt, wo sich die Personen befinden und was sie tun. Daraus folgt, dass dieser Simulator keinen Verkehr als solchen generiert, sondern Aktivitäten einer bestimmten Anzahl von Personen (Population) in einer bestimmten Region (Netzwerk) für einen bestimmten Zeitraum. Der Verkehr wird durch diese Personen induziert. Das entspricht auch der Realität, denn der Verkehr ist nicht der Selbstzweck, sondern ein Mittel zum Zweck.

- Auf solche Weise benötigt man im ersten Schritt für ein neues Szenario ein Netzwerk. Als Grundlage können dazu unterschiedlichste Daten dienen (z.B. GIS, Karten, Pläne, GoogleEarth). Diese Daten müssen in Form von Koordinaten vorliegen oder in eine solche umgewandelt werden, damit der Simulator sie verwenden kann, denn er ist so konzipiert, dass alle Aktivitäten der synthetischen Population entlang der Kanten stattfinden. Dafür ist in MATSim das Tool „org.matsim.run.XY2Links“ integriert, das jeder Aktivität aufgrund ihrer Koordinaten eine Kante zuordnet. So setzt sich ein Netzwerk aus Knoten „nodes“ und Kanten „links“ zusammen. Knoten müssen durch eine Identifikationsnummer „id“ sowie die Koordinaten „x, y“ definiert werden. Die Kanten werden durch die „id“, den Anfangs- und den Endknoten, die Länge „length“ und die Kapazität „capacity“ bestimmt, außerdem können beliebige andere Parameter, solche wie z.B. „freespeed“ oder „oneway“, angegeben werden, mit denen konkrete Kanten charakterisiert und beschrieben werden. Dabei besteht eine Möglichkeit, die Eigenschaften der Strecken variabel in der Zeit zu gestalten. Auf solche Weise lassen sich verschiedene Ausnahmesituationen programmieren, wie z.B. Unfälle, Evakuationen. Ein Beispiel für ein Netzwerk in der Form einer XML-Datei ist in Abbildung 8 zu sehen:

```

- <network>
  - <nodes>
    <node id="1" x="0" y="0"/>
    <node id="2" x="40" y="0"/>
    <node id="3" x="60" y="0"/>
    <node id="4" x="100" y="0"/>
  </nodes>
  - <links capperiod="01:00:00">
    <link id="1" from="1" to="2" length="40" freespeed="20" capacity="1000.0" permlanes="1"/>
    <link id="2" from="2" to="3" length="20" freespeed="20" capacity="1000.0" permlanes="1"/>
    <link id="3" from="3" to="4" length="40" freespeed="20" capacity="1000.0" permlanes="1"/>
    <link id="11" from="4" to="3" length="40" freespeed="20" capacity="1000.0" permlanes="1"/>
    <link id="12" from="3" to="2" length="20" freespeed="20" capacity="1000.0" permlanes="1"/>
    <link id="13" from="2" to="1" length="40" freespeed="20" capacity="1000.0" permlanes="1"/>
  </links>
</network>

```

Abbildung 8: Beispiel eines Netzwerkes mit vier Knoten und sechs gerichteten Kanten/drei Verbindungen.

Quelle: <http://matsim.org/uploads/xy2links/network.xml>

- Im zweiten Schritt der Bildung eines Szenarios muss die synthetische Population der Personen generiert werden. Jeder einzelne Agent braucht dabei eine Identifikationsnummer „id“ und zumindest einen Plan „plan“, der seine Aktivitäten über die ganze Zeitperiode kontinuierlich abbildet. Jeder Plan enthält somit Aktivitäten „acts“ und Verkehrsmittel „legs“. Jede Aktivität muss durch „type“ und Koordinaten „x, y“ definiert werden. Die erste Aktivität benötigt außerdem die Information „end_time“, alle weiteren außer der letzten die Dauer „dur“. Beide Zeitangaben müssen keine exakten Werte sein, da sie i.d.R. im Laufe der Simulation optimiert⁵ und an die Randbedingungen der Simulation unter der Berücksichtigung der Pläne anderer Agenten angepasst werden. Die Verkehrsmittel werden durch „mode“ beschrieben. Ein Beispiel dazu ist in Abbildung 9 zu sehen:

```

- <plans>
  - <person id="1">
    - <plan>
      <act type="home" x="20" y="-5" end_time="06:00:00"/>
      <leg mode="car"/>
      <act type="work" x="80" y="5" dur="08:00:00"/>
      <leg mode="car"/>
      <act type="home" x="20" y="-5"/>
    </plan>
  </person>
</plans>

```

Abbildung 9: Beispiel für die Beschreibung einer Person.
Quelle: <http://matsim.org/uploads/xy2links/plans.xml>

Dabei kann jeder Agent mehrere Pläne besitzen. Zu Beginn der Simulationsausführung wird pro Agent jeweils ein Plan ausgewählt, als aktiv markiert und als Initialnachfrage des Agenten ausgeführt.

Es gibt keine Auflistung aller zur Wahl stehenden „types of acts“ und „modes of legs“, da sie abhängig von den Bedürfnissen der konkreten Simulation beliebig programmiert werden können. Zum Beispiel: Bei der Simulation der gesamten Verkehrsnachfrage der Schweiz, die bereits im Kapitel 1.1 erwähnt wurde und ausführlicher im Kapitel 1.3 und im Kapitel 2 betrachtet wird, wurden fünf „modes“ (und zwar: car driver, car passenger, bicycle, walk, public transport) und 17 „types“ (unter anderem: work_sector2, work_sektor3, kindergarten, primary school, secondary school, higher education, culture, sports usw.) generiert (vgl. Balmer et al. [5]).

⁵ Vgl. Balmer [6] und das Kapitel 1.2.3. Im Kapitel 3.2 wird anhand eines Beispiels ausführlicher auf die Module zur Optimierung der Routen („ReRoute“) und der Zeiten („TimeAllocationMutator“) eingegangen.

Unter Optimierung wird dabei der Versuch verstanden, die Verhaltensweise der Personen so realitätsnah wie möglich zu simulieren. Durch die Optimierung wird es z.B. möglich, die Entscheidung einer Gruppe von Personen zu simulieren, z.B. eher oder später zur Arbeit zu fahren oder andere Wege zu benutzen, wenn es auf der gewohnten und sonst bevorzugten Strecke zu den Staus oder Beeinträchtigungen kommen kann. Zur Optimierung der Pläne von Agenten können Variationen der Routen, Zeiten, sowohl der Verkehrsmittel, Orte und der gesamten Events-Ketten vorgenommen werden.

1.2.2 Events

Events der synthetischen Population nehmen als Aktivitäten den zentralen Platz in dem Konzept des Verkehrssimulators MATSim ein. Durch Kombination verschiedener Typen von Events kann ein beliebiger Tagesablauf der Personen simuliert werden. Zum aktuellen Zeitpunkt stehen im MATSim folgende Events zur Verfügung:

id	Bezeichnung	Daten	Funktion/Beschreibung
0	arrival	time, agent-id, link-id	Agent kommt an der Kante um eine bestimmte Zeit an.
2	left link	time, agent-id, link-id	Agent verlässt die Kante um eine bestimmte Zeit. Zusammen mit dem Event 5 bildet dieses eine logische Gruppe: jeder Agent bewegt sich durch das Netz, indem er bei einer Kante ankommt und sie dann verlässt, bevor er seine Zielkante erreicht.
3	stuckAndAbort	time, agent-id	Event wird präventiv zum Auflösen von Staus genutzt, indem der „festsitzende“ und dadurch störende Agent entfernt wird, damit die anderen sich fortbewegen können.
4	wait2link	time, agent-id	Nachdem ein Agent eine Lokalität verlassen hat, muss er ggf. auf die Möglichkeit der Einfahrt in die Straße warten. Oft fällt dieses Event mit dem Event 6 zusammen, aber nicht immer.
5	entered link	time, agent-id, link-id	Zusätzlich zum Event 2 wird an den Knoten verwendet, um sie zu passieren.
6	departure	time, agent-id, link-id	Agent verlässt die Lokalität.
7	actstart	time, agent-id, link-id, act-type	Agent beginnt mit einer Aktivität.
8	actend	time, agent-id, link-id, act-type	Agent beendet eine Aktivität.
9	agentMoney	time, agent-id, amount	Agent erwirbt eine Summe, z.B. um eine Maut zu bezahlen.

Tabelle 1: Übersicht der Event-Typen.

Eine typische Kombination dieser Events wird in Abbildung 10 dargestellt und erläutert:

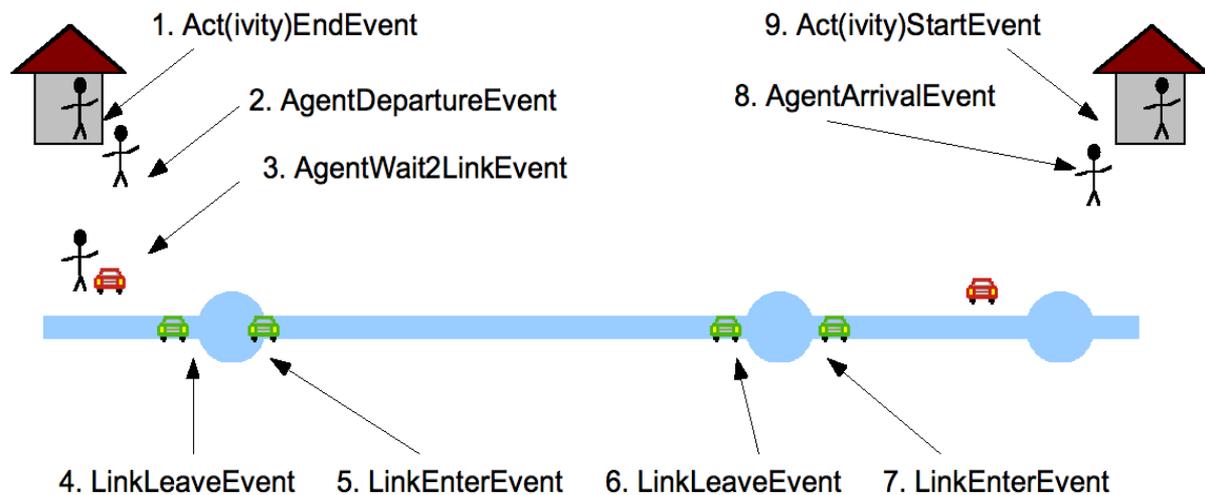


Abbildung 10: Events als zentrales Instrument der Simulation. Das abgebildete Beispiel wird mit der folgenden Event-Kette beschrieben:

Event 1: actend home (Ende einer Aktivität zu Hause) -> Event 2: departure (Verlassen einer Lokalität, des Hauses) -> Event 3: wait2link (Warten auf die Möglichkeit des Einbiegens in die Hauptstraße) -> Event 4 & Event 5: left link&entered (Überwinden eines Knotenpunktes⁶) -> Event 6: arrival (Ankunft am Ziel) -> Event 7: actstart (Beginn einer Aktivität am Ziel).

Quelle: http://matsim.org/uploads/18/eventsoverview_png_49f05de8ee.png

Es besteht auch die Möglichkeit, neue Events zu generieren. Außerdem wurde von den Entwicklern die Notwendigkeit neuer Events bereits erkannt, da die zum heutigen Zeitpunkt zur Verfügung stehenden Events bei weitem nicht alle Simulationsbedürfnisse abdecken (z.B. Miteinbeziehung des öffentlichen Personennahverkehrs ÖPNV). In Aussicht wurden folgende Event-Typen gestellt:

Event Name	Attributes
EndActivity	person, facility, actType
EnterVehicle	person, vehicle, facility
DepartureAtFacility	vehicle, facility
FacilityToLink	vehicle, link
LinkToIntersection	vehicle, link
IntersectionToLink	vehicle, link
LinkToFacility	vehicle, link
ArrivalAtFacility	vehicle, facility
LeaveVehicle	person, vehicle, facility
StartActivity	person, facility, actType
VehicleStuck	vehicle, link

Abbildung 11: Zukünftige Event-Typen bei MATSim.

⁶ Simulation der Knotenpunkte wird im Kapitel 1.3 näher betrachtet.

1.2.3 Module

Der modulare Aufbau des Verkehrssimulators MATSim erlaubt es, einzelne Module separat zu nutzen und sie nach Bedarf sowie dem optimalen Verhältnis zwischen Funktionalität, Detaillierungsgrad und Laufzeit/Speicherplatz beliebig zu kombinieren. Balmer et al. [5] empfehlen für die Generierung der Simulationen die in Abbildung 12 dargestellte Konfiguration der Module. Gleichzeitig sie weisen aber darauf hin, dass die Module in verschiedenen Anwendungen oft auch in der gleichen Konfiguration unterschiedliche Gewichtung bekommen können. Außerdem ist es möglich und von den Projektentwicklern erwünscht, in den Verkehrssimulator MATSim neue Module zu implementieren, um dadurch seine Fähigkeiten zu erweitern und seine Leistung zu steigern.

Table 2 Scheduling: Current preferred configuration in MATSim-T

Task	Frequency per run	Model type	Reference
Number, sequence and type of activities	Once	Conditional probability	Hettinger, 2007
Start and duration of activities	Per iteration	Best response model (GA-based optimizer)	Meister <i>et al.</i> , 2006
Composition of the group undertaking the activity	---	---	
Expenditure and its allocation among the participants	---	---	
Secondary location choice	Once	Imputed (Proportional to size and distance)	
Mode/vehicle choice	Initial iteration	Imputed (Chain based MNL)	Ciari <i>et al.</i> , 2007
Mode choice (car, public transport) ¹	Per iteration	Imputed (MNL based on plan score)	
Route choice	Per iteration	Best response model (A* landmark shortest path)	Lefebvre and Balmer, 2007
Point of egress from the vehicle	---	---	
Composition of the group travelling together	---	---	
Travel expenditure and its allocation among the travellers	---	---	

¹ The public transport candidate plan is generated as part of the initial demand generation for all agents which have a car available for travel.

Abbildung 12: Empfohlene Konfiguration der Module bei der Generierung der Simulationen mit MATSim.

Quelle: Balmer et al. [6], <http://matsim.org/uploads/BalmerEtAl2008ITM.pdf>

Zu bemerken ist außerdem, dass das Modul „route choice“ momentan nur für die Simulationen mit dem „mode“ = „car“ zur Verfügung steht. Desweiteren werden die Agenten, die mit dem „mode“ = „public transport“ unterwegs sind, in einem Schritt von ihrem Ausgangspunkt zu ihrem Ziel bewegt. Auch das soll sich demnächst ändern.

1.3 Grenzen und Realitätsnähe des Modells

Wie im Kapitel 1.1 bereits erwähnt wurde, ist MATSim im Stande, riesige⁷ Szenarien zu simulieren. Daher ist es naheliegend, die Grenzen des Programms nicht in der Anzahl der simulierten Personen (Agenten) oder in der Größe der verwendeten Netzwerke, sondern zum einen in dem Detaillierungsgrad der Realitätsabbildungen sowie der Realitätsnähe und zum anderen in den dafür benötigten Rechnerkapazitäten⁸ zu suchen.

Daraus ergeben sich folgende zu untersuchende Punkte:

1. Was kann mit dem Verkehrssimulator abgebildet werden und was nicht? Welche Module stehen in dem aktuellen Entwicklungsstadiums des Programms dafür zur Verfügung?
2. Wie aufwändig ist die Implementierung verschiedener Objekte?
3. Wie realitätsnah ist dann die Simulation? Stimmt sie mit den statistischen Erhebungen überein?

Eine detaillierte und alle Bereiche umfassende Untersuchung der genannten Punkte würde den zeitlichen Rahmen dieser Arbeit sprengen. Daher können an dieser Stelle nur beispielhaft einige ausgewählte Aspekte betrachtet werden.

Als erstes wird darauf hingewiesen, dass sich der Verkehrssimulator MATSim in einem Entwicklungsstadium befindet. Die Entwickler hoffen, in etwa drei bis vier Jahren ein vollständiges Programm auf den Markt bringen zu können. Momentan sehen sie unter anderem folgende Verbesserungsmöglichkeiten (vgl. Balmer [5]):

- Minimierung der Laufzeiten und des Speicherplatzbedarfs für größere Simulationen (Größenordnung des Beispiels „Verkehrsnachfrage der Schweiz“⁹), unter anderem auch durch Reduzierung der Anzahl der bis zum Erreichen des optimalen Zustandes benötigten Iterationen¹⁰.
- Implementierung eines Moduls für Subtouren, um kürzere Wege - wie z.B. vom Arbeitsplatz zur Kantine und zurück - realistischer abbilden zu können, ohne auf die Hauptressourcen des Rechners zugreifen zu müssen.
- Verfeinerung der Entscheidungskriterien für die Auswahl der optimalen Kombination von Aktivitäten in den Plänen der Agenten (die Wahl des Verkehrsmittels kann beispielsweise durch den Zoll, Parkgebühren, Mitfahrgelegenheit beeinflusst werden).

⁷ Neumann [1]: „MATSim ist in der Lage, Szenarios mit bis zu 8 Millionen Agenten gleichzeitig zu simulieren und dabei Netze zu verwenden, die mehrere hunderttausend Kanten umfassen und beispielsweise die gesamte Schweiz abdecken“.

⁸ Siehe auch Kapitel 2 „Laufzeit- und Speicherplatzuntersuchungen“.

⁹ Siehe auch Kapitel 2 „Laufzeit- und Speicherplatzuntersuchungen“.

¹⁰ Siehe auch den Anhang A.

- Implementierung neuer Events zur detaillierteren Beschreibung der Tagesabläufe der Agenten (vgl. Kapitel 1.2.2).
- Weitere Optimierung des zentralen Moduls „Verkehrssimulation“, z.B. durch die Gestaltung des Moduls als multi-modales Tool, um den Agenten die Möglichkeit zu geben, in einem konsistenten Netzwerk reibungslos die Verkehrsmittel („modes“) zu wechseln, darunter auch das Miteinbeziehen des ÖPNV.
- Erweiterung des Verkehrssimulators durch neue Module mit dem Ziel, seine Leistungsfähigkeit zu steigern.

Mehrere Entwickler arbeiten derzeit an verschiedenen Modulen. Als Beispiel für einen der letzten wesentlichen Beiträge kann die Diplomarbeit von Neumann genannt werden. Neumann widmet sich der Weiterentwicklung des im MATSim verwendeten Warteschlangenmodells mit dem Ziel, Lichtsignalanlagen (LSA) und ihre Auswirkungen auf den Verkehrsfluss abbilden zu können. Nach seiner Beurteilung ist das Modell jetzt *„in der Lage, die für die Kapazität eines Knotenpunktes entscheidenden Eigenschaften wie Anzahl der zur Verfügung stehenden Fahrstreifen, deren Abbiegebeziehungen und Länge abzubilden. Es ist möglich, an einem Knotenarm für jede Abbiegebeziehung getrennte Fahrstreifen einzurichten, mehrere Abbiegebeziehungen auf einem gemischten Fahrstreifen zusammenzufassen oder beides zu kombinieren. Zusätzlich kann die Signalisierung unabhängig von der Aufteilung der Fahrstreifen erfolgen. So können mehrere Fahrstreifen von einer Signalgruppe gemeinsam signalisiert, einzelne Abbiegebeziehungen eines gemischten Fahrstreifens von separaten Signalgruppen signalisiert werden oder es wird auf eine Signalisierung der Spuraufteilung verzichtet. Rückstaus auf einem Fahrstreifen blockieren ungestaute Fahrstreifen und breiten sich auf den gesamten Knotenarm und stromaufwärts anschließende Kanten aus“* (Neumann [1]). Eine solche Erweiterung verbessert die Leistungsfähigkeit und die Realitätsnähe des Modells erheblich, denn davor konnten LSA nur dadurch berücksichtigt werden, dass man sie durch Absenken der Kapazität auf der ganzen Strecke (beide Richtungen) imitiert hat. Das entsprach nicht der Realität der Staubildung nur auf den Stellen und Fahrstreifen, die tatsächlich von den LSA-Regelungen betroffen werden. Somit ist das Modell momentan in der Lage, einen beliebigen Knoten mit oder ohne LSA abzubilden. Dabei ist aber zu bemerken, dass es bislang kein Modul gibt, das den Verkehrsfluss im Knoten selbst abbilden kann. Kanten führen zum und vom Knoten, der Knoten selbst bleibt weiterhin eine „Blackbox“, vgl. Abbildung 13:

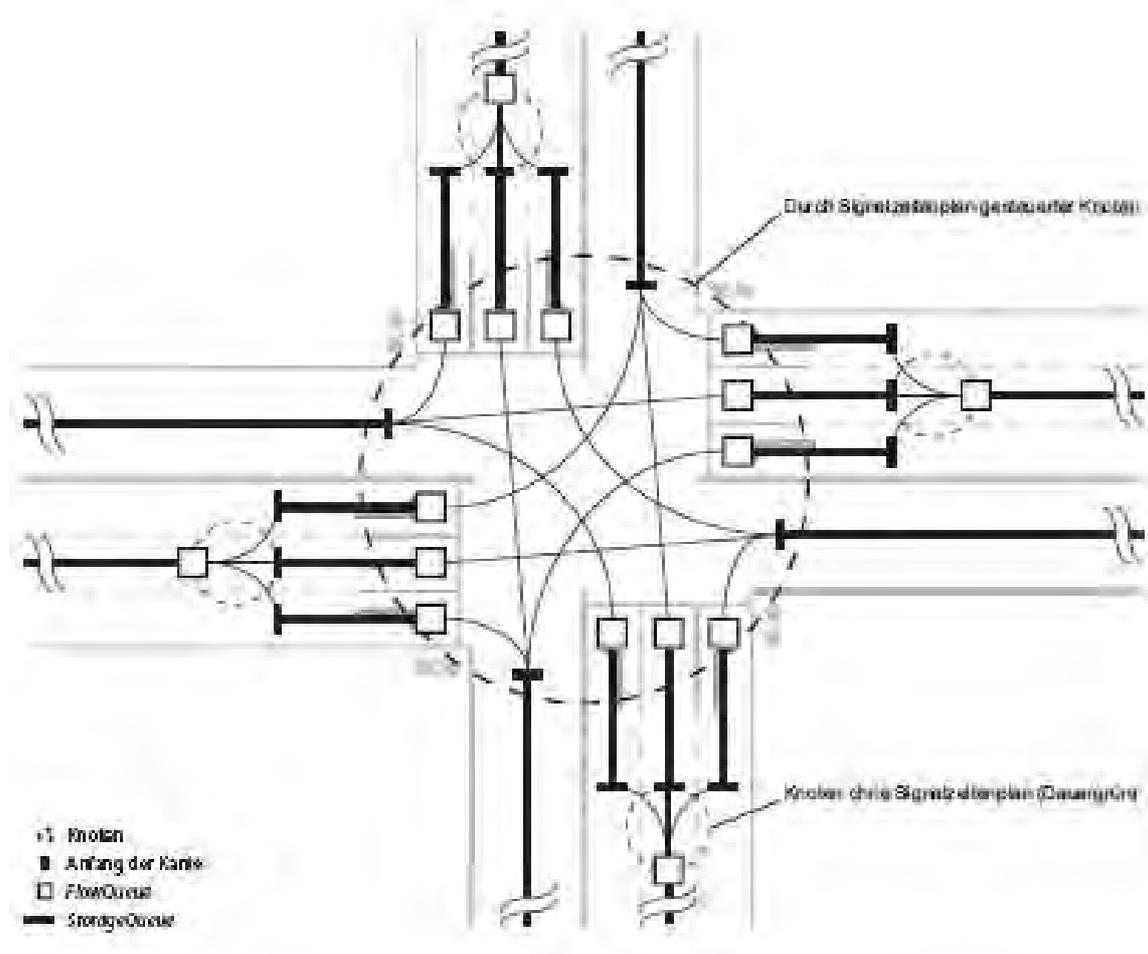


Abbildung 13: Implementierung eines vierarmigen Knotenpunktes im MATSim.

Quelle: Diplomarbeit von Neumann [1], <https://svn.vsp.tu-berlin.de/repos/public-svn/publications/vspwp/2008/08-24/Neumann2008Dipl.pdf>

Neumann hat auch den Aspekt der Implementierung des Wendevorgangs im MATSim untersucht. Er stellt fest, dass es oft keine Möglichkeit zum Wenden gegeben ist, denn „die Kanten in MATSim sind unidirektional. Somit ist das Wenden auf einer einzelnen Kante nicht möglich, sondern kann immer nur an einem Knoten erfolgen. Deshalb wird Wenden hier als der Wechsel zwischen zwei entgegengesetzt und parallel verlaufenden Kanten definiert. Besitzt eine Kante eine zu ihr entgegengesetzte Kante, so ist ein Wendevorgang möglich“ (Neumann [1]). Andernfalls ist das Wenden nicht möglich. In seiner Diplomarbeit erweitert Neumann das Modell so, dass jetzt die Aktivitäten beliebig auf einer Kante verankert werden können. Als einen der Vorteile nennt der Autor auch die Möglichkeit, die metergenau frei zu setzende Entscheidungspunkte nicht nur zum Abbilden der Wendevorgänge, sondern auch z.B. zur Implementierung des öffentlichen Personennahverkehrs (Haltestellen) zu nutzen. Der entscheidende Nachteil ist dabei der Aufwand der Implementierung, der mit jedem neuen Detail, d.h. mit dem steigenden Detaillierungsgrad und der Genauigkeit der Abbildung erheblich steigt.

Trotz vorhandener und oben genannter Verbesserungsmöglichkeiten hat sich MATSim in mehreren Projekten bereits als funktions- und leistungsfähig bewährt. Zu einigen dieser Projekte wurden auch Studien durchgeführt, die die Nähe der Simulationen zur Realität untersuchten. Die Literaturrecherche hat folgendes ergeben:

1. Projekt „Simulation der Verkehrsnachfrage der Schweiz“

Eine zusammenfassende Beurteilung der Realitätsnähe der Simulation findet sich in der Arbeit von Meister et al. [4] wieder: *„Es wurde gezeigt, dass agentenbasierte Simulation des Verkehrsverhaltens in der Lage ist, in großem Maßstab den dynamischen Zustand des Straßenverkehrs der Schweiz abzubilden. Die Verkehrsnachfrage kann in ihrer tageszeitlichen Dynamik gut abgebildet werden, was ein Vergleich der echten und simulierten Verkehrszählungen zeigt. Die Nachfrageerzeugung beschränkt sich auf den inländischen Personenverkehr und lässt Transit- sowie Werksverkehr außen vor. Deshalb sind die Verkehrsmengen am Rand des modellierten Gebietes bzw. auf den Haupttransitachsen zu niedrig, abgesehen davon werden die Volumina in der richtigen Größenordnung reproduziert. Die simulierten Wegreisezeiten stimmen mit entsprechenden Beobachtungen überein. Die iterierte Reisezeit von 7 min, 40 sec ist gut mit dem Wert aus dem Mikrozensus vergleichbar. Teilt man die beobachtete Unterwegszeit mit dem Auto an Werktagen (27,3 min) durch die beobachtete Anzahl Wege pro Werktag (3,6) ergeben sich ebenso exakt 7 min, 40 sec. Diese exakte Übereinstimmung ist sicherlich zufällig und die Berechnungsweise naiv: In der Realität werden nur ca. 1,2 Wege pro Person und Tag mit dem Auto zurückgelegt. Dieser Wert bezieht sich jedoch auf die gesamte Schweizer Bevölkerung von ca. $7,3 \cdot 10^6$ Personen, während wir nur $2,3 \cdot 10^6$ Agenten simulieren. Der Vergleich zeigt aber, dass das System Resultate in der richtigen Größenordnung erzeugt. Wird das Modell der Verkehrsmittelwahl wie geplant auf die Ebene von Subtouren verfeinert, wird sich die simulierte durchschnittliche Unterwegszeit mit dem Auto erhöhen, wird sich die Anzahl Wege/Tag die mit dem Auto zurückgelegt werden verringern.“*

Das Balkendiagramm aus der Arbeit von Balmer et al. [5] in Abbildung 14 veranschaulicht die Übereinstimmung der Ergebnisse differenziert nach der Uhrzeit:

Figure 4 Example comparison between simulated and counted traffic flows

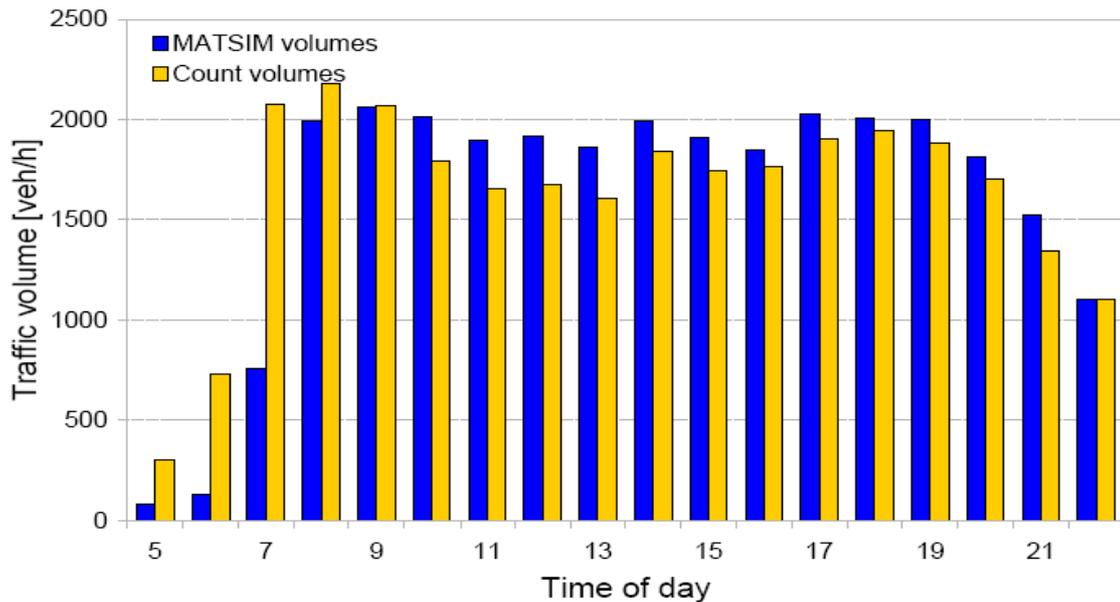


Abbildung 14: Vergleich der Simulationsergebnisse der schweizerischen Verkehrsnachfrage (blaue Balken) mit den Daten der Zählstellen (gelbe Balken).

Quelle: Balmer [5], <http://matsim.org/uploads/BalmerEtAl2008ITM.pdf>

Als Gründe für die Abweichung nennt Horni [7] Folgende (Vgl. Abbildung 15):

- Durchgangsverkehr konnte nicht berücksichtigt werden und
- Wirtschaftsverkehr (z.B. Taxis) wurde nicht genau erfasst.



Abbildung 15: Gründe für die Abweichungen zwischen den Simulationsergebnissen und den Daten aus den Zählstellen im Projekt „Simulation der Verkehrsnachfrage der Schweiz“.

Quelle: Horni [7], http://matsim.org/uploads/ScienceCity2007_Horni.pdf

2. Projekte „Durchquerung des Baregg-Tunnels“ und „Autobahn A3, Höhe Adliswil“

Auch Balmer et al. [5] weisen darauf hin, dass die simulierten Werte generell zu tief liegen. In ihrer Arbeit gehen sie detailliert auf die Stärken und Schwächen der beiden Simulationen ein: *„Die Qualität der Nachfragemodellierung kann durch den Vergleich realer Zählzeiten mit künstlichen Verkehrszählungen der Simulation beurteilt werden. Von den zuständigen Bundes- und Kantonsbehörden stehen uns stundenfeine Zählzeiten eines ganzen Jahres zur Verfügung. Da das Ziel ist, das Verkehrsverhalten eines durchschnittlichen Werktages zu modellieren, wurden die Zählungen von Dienstag, Mittwoch und Donnerstag zu einem repräsentativen Tag zusammengefasst. Ferien und Feiertage wurden von der Aggregation ausgeschlossen. Die Vergleiche der Tagesganglinien am Baregg-Tunnel sowie einem Autobahnquerschnitt nahe Adliswil bei Zürich zeigen deutlich die Stärken und die Schwächen des Szenario. Die qualitative Struktur der jeweiligen Pendlerspitzen kann gut nachgebildet werden, was für die Repräsentativität der Zeitangaben im Mikrozensus, deren Verteilung auf die Agenten in der initialen Nachfragemodellierung und die Routenwahl während der Iterationen spricht. Während jedoch im Falle Adliswil auch die Volumina gut getroffen werden, gibt es am Baregg in der Simulation nur halb so viel Verkehr wie in der Realität. Wir vermuten, dass der in der agentenbasierten Nachfrage fehlende Transit- und Werksverkehr für diese Lücken verantwortlich ist. In den Verteilungen der Aktivitätszeiten, und deshalb auch in den Zählzeiten der Simulation zeigt sich jeweils eine kleine Mittagsspitze, welche durch den Weg zur und von der Mittagspause erzeugt werden. Diese Spitze ist in den erhobenen Zählzeiten nicht erkennbar.¹¹ Der Grund ist, dass der komplette Aktivitätenplan eines simulierten Agenten mit dem Auto absolviert wird, demzufolge auch diese mittäglichen Wege. In der Realität handelt es sich jedoch in der Regel um Fußwege. Deshalb wird auch dieser qualitative Fehler höchstwahrscheinlich durch das geplante tourenbasierte Modell der Verkehrsmittelwahl korrigiert werden. Eine Zusammenfassung der Zählzeitenvergleiche zeigt eine durchschnittliche Unterschätzung der Verkehrsnachfrage um 50 %.“*

¹¹ Vgl. den Verlauf der roten und grünen Balken zwischen 12:00 – 15:00 in den Abbildungen 16 und 17

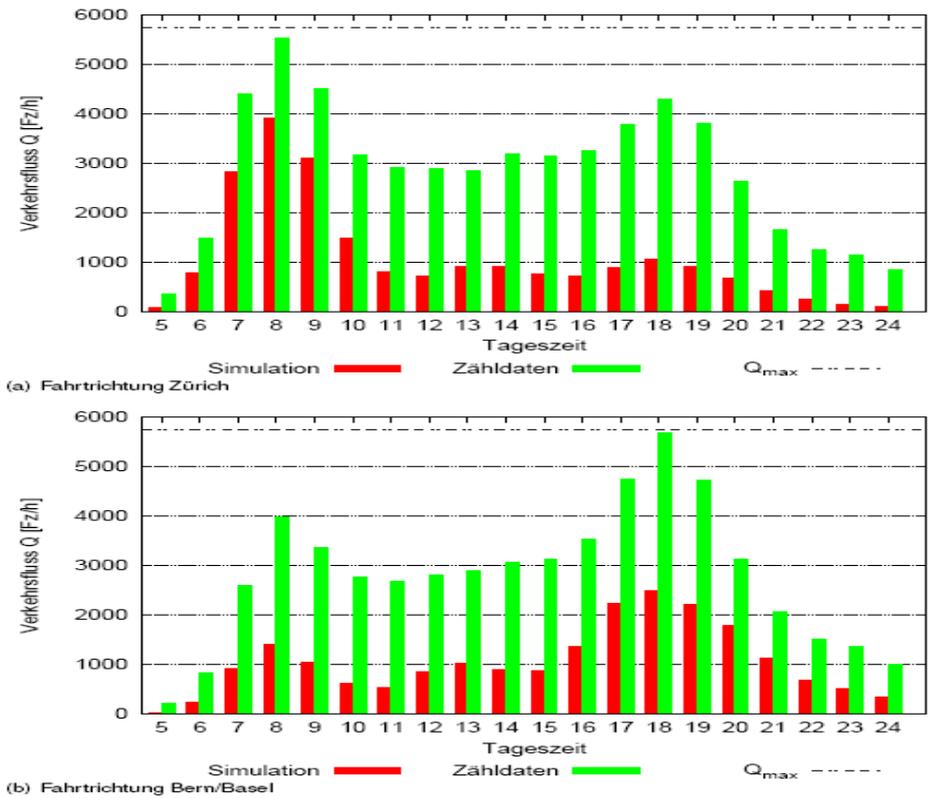


Abbildung 16: Vergleich Simulation - Zählstelle an der Autobahn A1, Baregg-Tunnel. Quelle: Balmer et al. [5], <http://matsim.org/uploads/BalmerEtAl2008ITM.pdf>

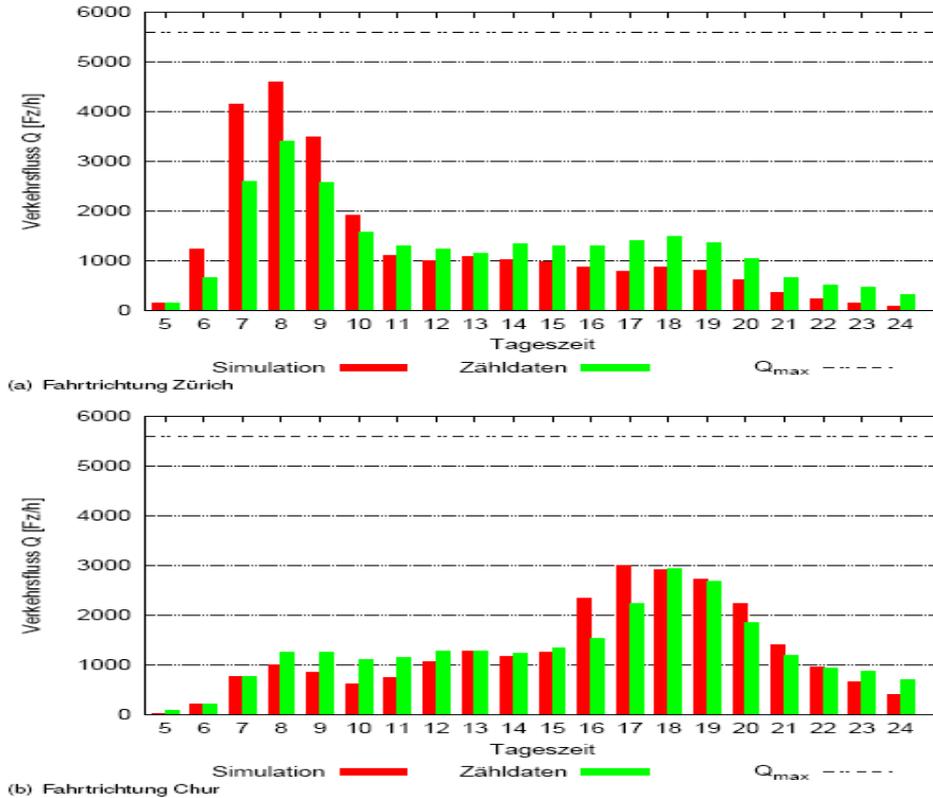


Abbildung 17: Vergleich Simulation - Zählstelle an der Autobahn A3, Höhe Adliswil. Quelle: Balmer et al. [5], <http://matsim.org/uploads/BalmerEtAl2008ITM.pdf>

2 Laufzeit- und Speicherplatzuntersuchungen

Der Dokumentation zum Verkehrssimulator MATSim kann Folgendes entnommen werden: Jeder Agent besitzt einige Eigenschaften (z.B. Geschlecht, Führerschein ja/nein, Auto ja/nein, Arbeitsverhältnis) und jedes Netzwerk wird durch die Aktivitäten, beschrieben mit dem „type“, und die Verkehrsmittel, beschrieben durch „mode“, charakterisiert (vgl. Kapitel 1.2.1). Diese Eigenschaften werden als Strings gespeichert. Jedes String, sogar ein leeres, beansprucht in Java bis zu 40 bytes und jedes Zeichen entspricht den 16 bit, so dass eben ein kleines String wie „f“ (Person.sex) oder „car“ (Leg.mode) viel Speicherplatz braucht. Bei der Generierung einer Population von 200 Personen mit 5 Eigenschaften werden dabei 1 Million Strings und mehr als 40MB RAM benötigt. Dieses Problem wurde bei MATSim durch die so genannten „object pools“ gelöst, in denen die typisierten üblichen Eigenschaften einmal abgespeichert werden und auf die andere Objekte zugreifen ohne eigene Eigenschaften vorzuhalten und separat speichern zu müssen. Dadurch reduziert sich die Anzahl der Strings auf 10 bis 15 Typen. Bei einer Simulation mit 160 Personen führte das zur Reduktion des Speicherplatzes um 25 %, wobei sich die Zeit für den Zugriff auf die Attribute kaum verändert hat (50s anstatt 49s). Es ist davon auszugehen, dass die Simulationen mit einer größeren Anzahl an Agenten eine noch effektivere Ersparnis des Speicherplatzes erzielt.

Da der Verkehrssimulator MATSim bereits für die Realisierung und Visualisierung einiger Projekte verwendet wurde, kann der Speicherplatz- und Rechenzeitbedarf unmittelbar anhand einiger Anwendungsbeispiele von MATSim veranschaulicht werden:

Erstes Anwendungsbeispiel

In der Arbeit von Meister et al. [4] wurde die Anwendung des Simulationssystems MATSim-T auf den Straßenverkehr der ganzen Schweiz präsentiert, auf die Untersuchung und Beurteilung des benötigten Speicherplatzes und vor allem der Rechenzeiten dieser Anwendung wurde sehr detailliert und ausführlich in der Arbeit von Balmer et al. [3] eingegangen. Zusammenfassend und nur um die ungefähren Dimensionen/Größenordnungen aufzuzeigen, wird in diese Bachelor-Arbeit Folgendes aus der oben genannten Veröffentlichung übernommen:¹²

¹² D.h. im folgenden Teil der Arbeit geht es um eine etwas abgeänderte Zitierung der genannten Veröffentlichung. Es wird dabei auf den ständigen Verweis auf die Arbeit von Balmer et al. [3] verzichtet.

Bei dieser Anwendung ging es um ca. 7,2 Mio. generierte Individuen (unter anderem mit demographischen Attributen wie Alter, Geschlecht, Führerscheinbesitz)¹³ für den motorisierten Individualverkehr auf etwa 7,1 Mio. Wege im nationalen Netzmodell mit 24.000 Knoten und ca. 60000 Kanten, die für einen durchschnittlichen Werktag auf einem Rechner mit acht Dual-Core Prozessoren und je 2,2 GHz Taktfrequenz vollständig zeit-dynamisch simuliert und optimiert wurden. Außerdem wurden etwa 1,7 Mio Gebäude modelliert, inklusive Informationen über Aktivitäten, die pro Gebäude ausgeführt werden können („types“: Heim, Arbeit, Schule, Einkauf, Freizeit). Der benötigte Arbeitsspeicher für diesen Anwendungsfall betrug 22 GByte. Dabei ist zu beachten, dass die Infrastrukturdaten (Gebäude, Netzwerk) verhältnismäßig wenig Speicher beansprucht haben, so dass beinahe der gesamte benötigte Speicherplatz für die Modellierung der Nachfrage gebraucht wurde. Die gesamte Rechenzeit belief sich auf 36 Stunden. Im Einzelnen:

a) Rechenzeit für die Simulation

Um den Verkehrsfluss abzubilden, werden zu jeder simulierten Sekunde des gesamten Tages Operationen ausgeführt. Demzufolge ist die Laufzeit proportional zu der Anzahl der Straßen des Verkehrssystems. Zu erwähnen ist, dass die Rechenzeit nicht von der Anzahl der Agenten im System abhängt. Dies ist günstig für ein stark ausgelastetes Netz, jedoch benötigt die Simulation die gleiche Rechenzeit für einen schwach ausgeprägten Nachtverkehr. Für die Simulation der gesamt-schweizerischen Nachfrage bedeutete dies 10.368 Mrd. Operationen mit der Gesamtrechenzeit von ca. 70 Minuten. Die Modellierung der individuellen, initialen Nachfrage für den Anwendungsfall ist detailliert in Meister et al. [2] zu finden. Die dazu verwendete Rechenzeit betrug ca. 6 Stunden.

b) Rechenzeit für die Optimierung und Überwindung der „Breakdown“-Situation

Der Begriff „Breakdown“ wird durch Balmer et al. [3] folgendermaßen erklärt: *„Durch den Optimierungsprozess und die gegebenen Randbedingungen (beispielsweise Zeitfenster für Beginn der Arbeit-Aktivität) kann es passieren, dass viele Agenten ähnliche Pläne ausprobieren, was zu hohen Verkehrsdichten auf bevorzugten Straßen führt. Entsteht dann an einer Stelle im Netzwerk ein Stau durch temporäre Überlastung, kann dieser Stau durch die allgemein hohe Verkehrsdichte nicht mehr aufgefangen werden. Es bilden sich Rückstaus, die sich immer weiter im Netzwerk ausbreiten. Die Auflösung dieser Staus dauert im Modell sehr lange, wodurch hohe Reisezeiten entstehen, was in der Nutzenfunktion zu großen negativen Beiträgen führt. Nach einem solchen Zusammenbruch des Netzwerks wechseln die meisten Agenten ihren Plan.“*¹⁴

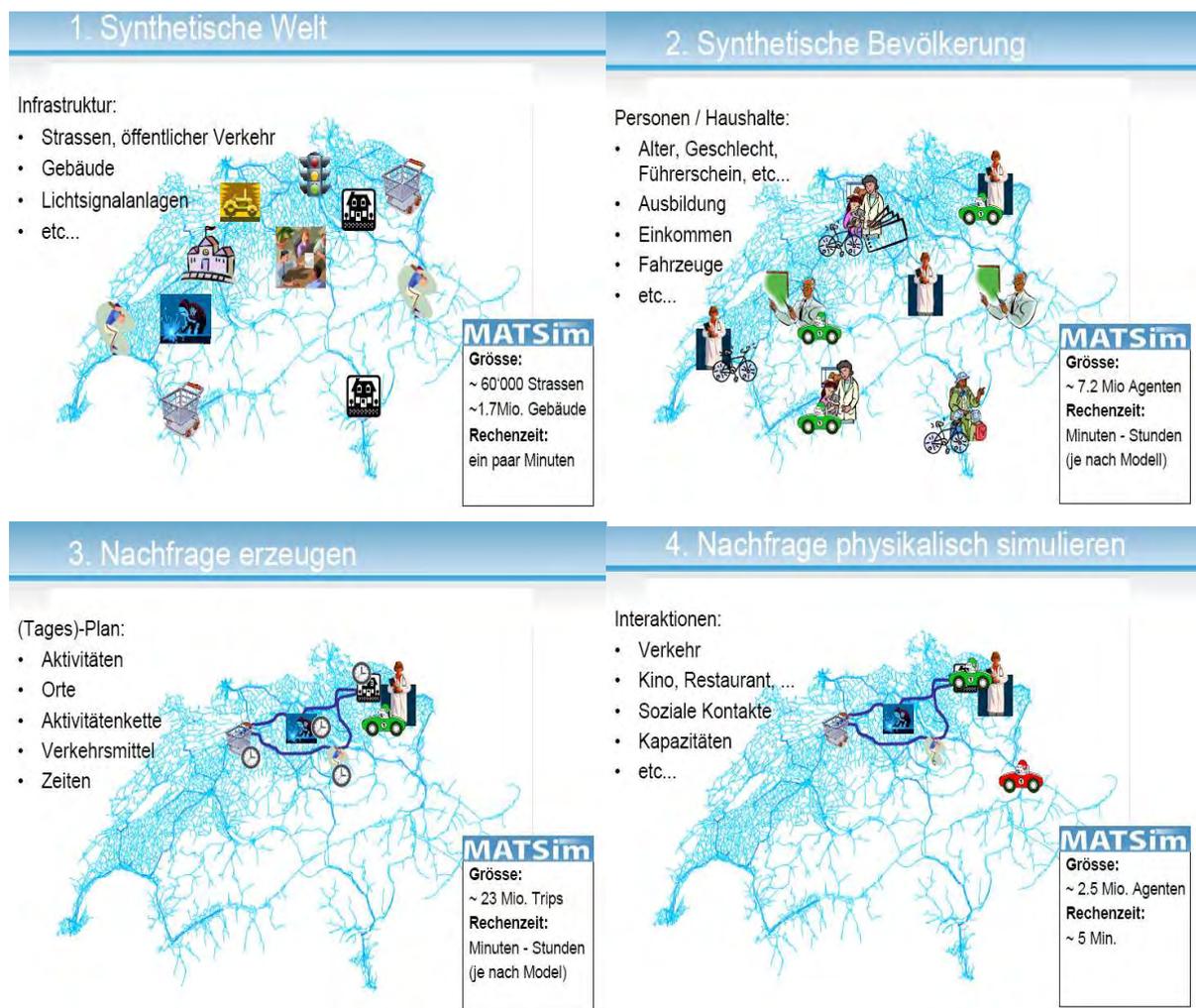
¹³ Eine vollständige Aufzählung der berücksichtigten Eigenschaften von Agenten findet sich auch in der Arbeit von Ciari et al. [8], und zwar: „Jeder Agent hat folgende Eigenschaften: Alter, Geschlecht, Arbeitsverhältnis (ja/nein), Nationalität (schweizerisch/nicht schweizerisch), Einkommen, Heimatadresse (hektargenau), Anzahl der Personen im Haushalt, Anzahl der Kleinkinder“.

¹⁴ Vgl. auch den Artikel von Scharf [9].

Eine Optimierungsiteration des Algorithmus von MATSim zur Berechnung der individuellen, zeit-dynamischen Gesamt-Tagesnachfrage für 7,1 Mio. Wege auf dem schweizerischen Planungsnetz mit 60.000 Kanten im Mittel etwa 32 Minuten. Mit der zusätzlichen Speicherung von Zwischenresultaten und Analysen von ca. 22 Minuten und dem Eintreten des optimalen Zustandes nach 100 Iterationen benötigte die Berechnung der resultierenden Nachfrage etwa 3,2 Tage.

Durch eine Re-Implementierung der parallelen DEQSim (Deterministische, event-basierte Warteschlangen-Simulation) in JAVA als integraler Bestandteil von MATSim-T ist es möglich, durch den Datenaustausch zwischen DEQSim und MATSim-T pro Iteration ca. 20 Minuten für den geschilderten Anwendungsfall einzusparen, was eine Reduktion von ca. 40 % des Gesamtaufwandes bedeutet. Ein Szenario der Größenordnung der gesamten Schweiz lässt sich somit in etwa einem Tag berechnen.

Im Rahmen einer Präsentation hat Horni [7] die Aufteilung der gesamten Rechenzeit zwischen den einzelnen Schritten sehr anschaulich in 6 Bildern dargestellt:



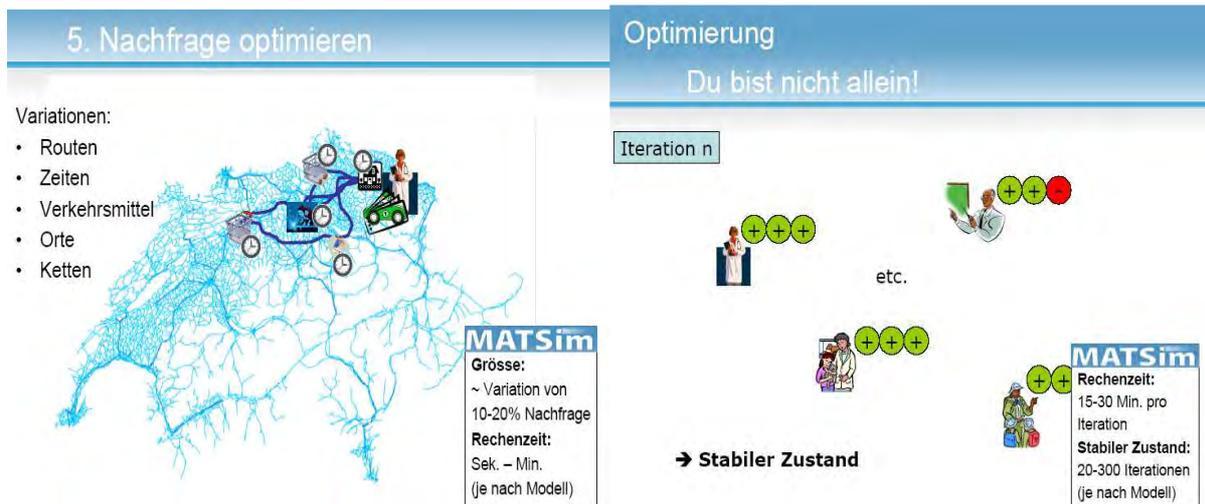


Abbildung 18: Aufteilung der gesamten Rechenzeit auf die einzelnen Schritte des Simulationsprozesses.

Quelle: Horni [7], http://matsim.org/uploads/ScienceCity2007_Horni.pdf

Noch zwei zusammenfassende Abbildungen zu diesem Anwendungsfall hinsichtlich der benötigten Rechenzeit findet man in der Arbeit von Balmer et al. [5]:

Table 4 Example scenario: Computing times

Processing step	Total for scenario [h]	Unit	Speed [Units/sec]	Frequency per run
Generating agents (CH)	0.12	Agent	20'000	Once
Scheduling (Fixed components) (CH)	14.40	Agent	140	Once
Filter 1.9 million agents (Zurich area)	0.30			Once
Filter 10% agents of Zurich area	0.10			Once
Scheduling (without routing)	0.04	Agent	100	Per iteration ¹
Scheduling (routing)	0.00	Agent	1'000	Per iteration ¹
Time-step traffic flow simulation	0.15	Agent	300	Per iteration
Learning/plan selection	0.00	Agent	250'000	Per iteration
Total for one iteration ¹	0.22			Per iteration
Total for run (for 100 iterations) ²	23.00			

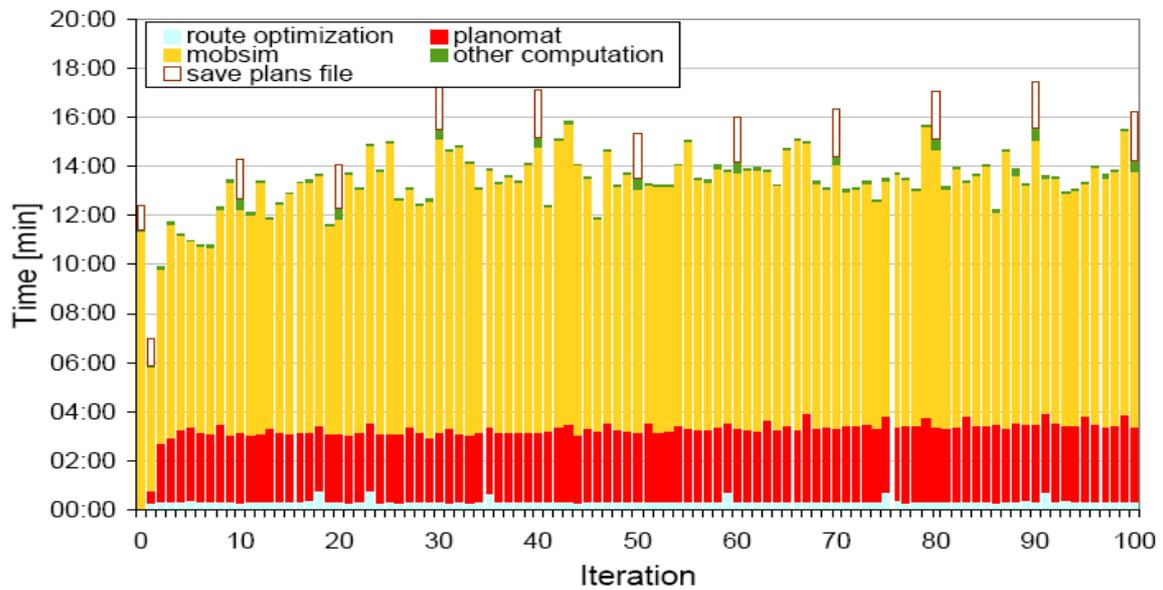
¹ Including various I/O operations; ² without the initial demand generation

Configuration: Time-step traffic flow simulation (mobsim) runs on a single CPU, replanning (router, planomat – time optimisation) runs in two threads; Dual-core AMD Opteron 2,6 GHz, 8 GB RAM

Abbildung 19: Rechenzeiten pro Prozessschritt.

Quelle: Balmer et al. [5], <http://matsim.org/uploads/BalmerEtAl2008ITM.pdf>

Figure 2 Computing times by step for the Greater Zürich scenario with 188'000 agents



Time-step traffic flow simulation (mobsim) runs on a single CPU, replanning (router, planomat – timing optimisation) runs in two threads; Dual-core AMD Opteron 2,6 GHz, 8 GB RAM

Abbildung 20: Rechenzeiten nach Modulen.

Quelle: Balmer et al. [5], <http://matsim.org/uploads/BalmerEtAl2008ITM.pdf>

Zweites Anwendungsbeispiel

In der Diplomarbeit von Meister [10] findet sich eine weitere Rechenzeitanalyse für folgende drei mit MATSim generierte und ausgeführte Szenarien:

1) Der Student.

In diesem Szenario übt der Student, der einen eigenen Haushalt in der Großstadt führt, folgende Aktivitäten aus: schlafen, aufstehen, arbeiten, zu Mittag essen, eine Arbeit präsentieren, einkaufen, ausgehen, zu Bett gehen. Die entstehenden Wege legt er mit einem Fahrrad zurück. Meister ließ 200000 Iterationen ablaufen und stellte das Ergebnis in zwei Diagrammen zusammen, die in dieser Arbeit in Abbildung 21 zu sehen sind:

Abbildung 3.1: Szenario Student: Legende



Abbildung 3.3: Szenario Student: Facility

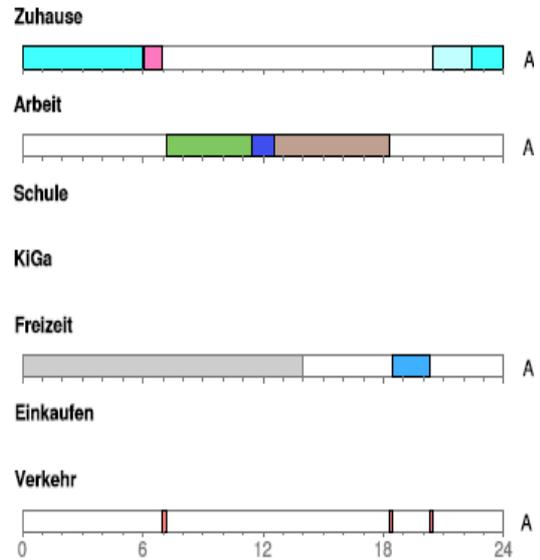


Abbildung 21: Ergebnis für das Szenario „Student“.
 Quelle: Diplomarbeit von Meister [10], <http://matsim.org/uploads/Meister2004master.pdf>

2) Zwei Arbeiter mit einem Kleinkind. Zwei Berufstätige, die in einem Vorort wohnen und ein Auto besitzen, üben teilweise gemeinsam folgende Aktivitäten aus: schlafen, aufstehen, Kind zur Kindertagesstätte bringen/holen, einkaufen, arbeiten, zu Mittag essen, ausspannen, zu Bett gehen. Das Ergebnis ist in Abbildung 22 zu sehen:

Abbildung 3.7: Szenario Zwei Arbeiter: Legende



Abbildung 3.9: Szenario Zwei Arbeiter: Faci

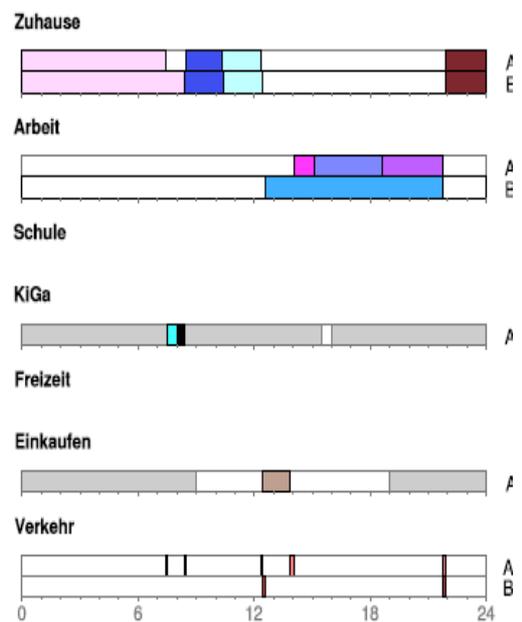


Abbildung 22: Ergebnis für das Szenario „Zwei Arbeiter mit einem Kind“.
 Quelle: Diplomarbeit von Meister [10], <http://matsim.org/uploads/Meister2004master.pdf>

3) Kleinfamilie mit einem Schulkind.

In diesem Szenario ist nur eine Person berufstätig. Die ausgeübten Aktivitäten je Person (A = berufstätig, B = nicht berufstätig, C = Schulkind), sowie der beste Plan nach der 600000. Iteration sind in Abbildung 23 zu finden:

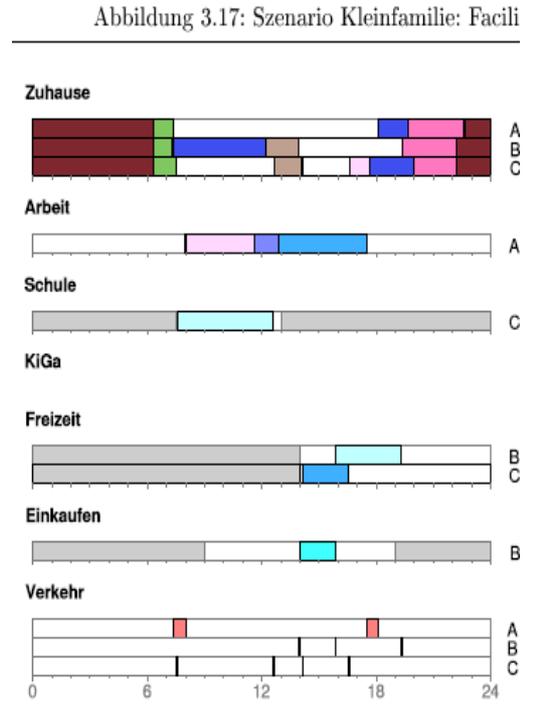
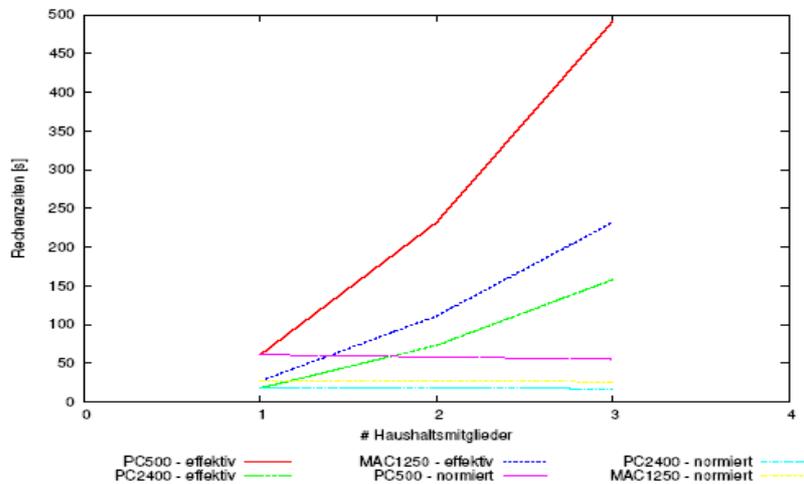


Abbildung 23: Ergebnis für das Szenario „Kleinfamilie mit einem Schulkind“. Quelle: Diplomarbeit von Meister [10], <http://matsim.org/uploads/Meister2004master.pdf>

Meister untersuchte in seiner Arbeit die Möglichkeit der Abstimmung der Aktivitäten innerhalb eines Haushaltes untereinander (unter anderem durch die optimale Wahl der Anfangszeiten für die Aktivitäten oder die Wahl des Verkehrsmittels) und führte für diese drei Szenarien auch die Analyse der dafür benötigten Rechenzeit durch. Das Ergebnis hat er folgendermaßen zusammengefasst: „Eine Analyse der benötigten Simulationszeit der drei Beispielszenarien auf drei verschiedenen schnellen Rechnern ist in Abbildung 4.1 dargestellt. Das fixe Abbruchkriterium (200.000 Generationen) und mit steigender Anzahl der Haushaltsmitglieder anwachsende Datenmenge lassen in Kombination die effektiven Rechenzeiten quadratisch ansteigen. Sie bewegen sich im Minutenbereich. Die normierte Rechenzeit dagegen ist die benötigte Dauer für einen Agenten bei einer angenommenen mittleren Generationenzahl von 200.000 durch ein geeignetes variables Abbruchkriterium. Hier zeigt sich erwartungsgemäß eine leicht abfallende Kurve, da bei großen (personenreichen) Szenarien der Aufwand für die fixen Simulationskomponenten hinter den der variablen Teile zurücktritt. Auf dem schnellsten Rechner (PC2400) benötigt die Simulation eines Agenten demnach durchschnittlich ca. 18 Sekunden. Bei diesem Modell ist in erster Linie die Taktfrequenz für die Berechnungsgeschwindigkeit entscheidend. Da

während der Simulation nicht sehr viele Daten geladen oder erzeugt werden müssen, spielt die Größe des Hauptspeichers eine untergeordnete Rolle“. (Meister [10]).

Abbildung 4.1: Rechenzeiten



PC500 Pentium III 500 MHz, 256 MB RAM

PC2400 Pentium IV 2,4 GHz, 1 GB RAM

MAC1250 Apple PowerBook, PowerPC G4 1,25 GHz Mobile, 1 GB RAM

Abbildung 24: Ergebnis für die Analyse der Rechenzeiten für alle drei Szenarien. Abbildung 4.1 aus der Diplomarbeit von Meister.

Quelle: Meister [10], <http://matsim.org/uploads/Meister2004master.pdf>

Zusammenfassend ist zu betonen, dass der meiste Speicherplatz für die Modellierung der initialen Nachfrage gebraucht wird, die längeren Rechenzeiten sind dagegen vor allem auf die benötigten Iterationen zurückzuführen. Im Allgemeinen gilt: Je detaillierter die initiale Nachfrage modelliert wurde, desto weniger Iterationen sind nötig, um in der Simulation den stabilen (optimalen)¹⁵ Zustand zu erreichen.

¹⁵ Siehe Anhang A.

3 Technische Details zum Verkehrssimulator MATSim

3.1. Installation des Simulators

Der Verkehrssimulator MATSim ist als OpenSource-Software im Internet verfügbar und benötigt keine Installation an einem bestimmten Computer. Um das Programm benutzen zu können, geht man folgendermaßen vor:

Auf der Homepage des MATSim gibt es oben rechts einen Button zum schnellen und unkomplizierten Download des Programms:

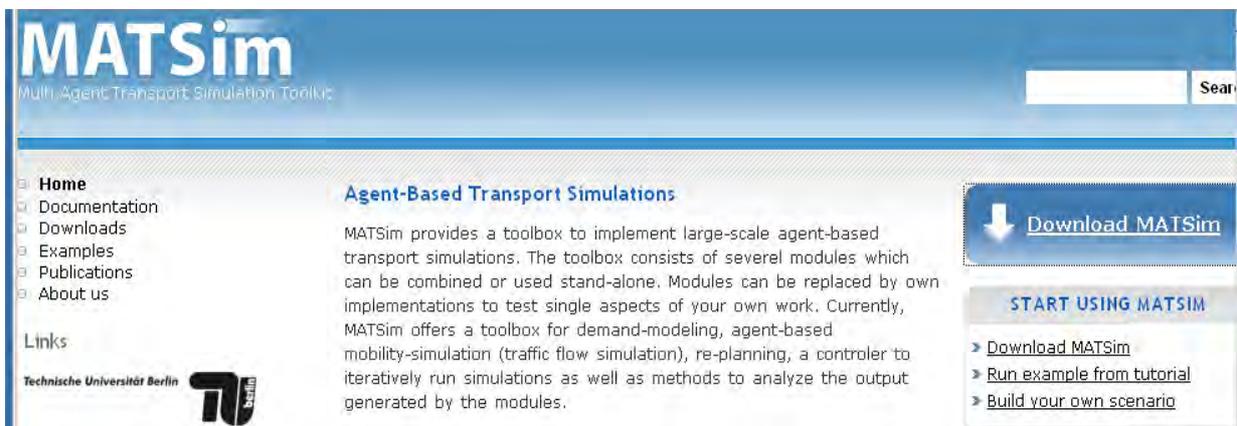


Abbildung 25: Button „Download MATSim“.

Auf der Seite sourceforge.net, auf die man automatisch weitergeleitet wird, steht die letzte, aktuelle Version des Simulators zum Herunterladen zur Verfügung:



Abbildung 26: Herunterladen des MATSim von der Seite sourceforge.net.

Im nächsten Schritt wird das Programm auf ein Speichermedium oder auf die Festplatte gespeichert. Damit ist der Download-Prozess abgeschlossen.

Um den Simulator starten zu können, geht man über „Start“ -> „Ausführen“ in die Befehlszeile. Nach der Eingabe „cmd“ gelangt man in das Eingabeaufforderungsfenster, wo

jedes File durch seinen Namen aus dem Verzeichnis der heruntergeladenen Dateien aufgerufen und ausgeführt werden kann.

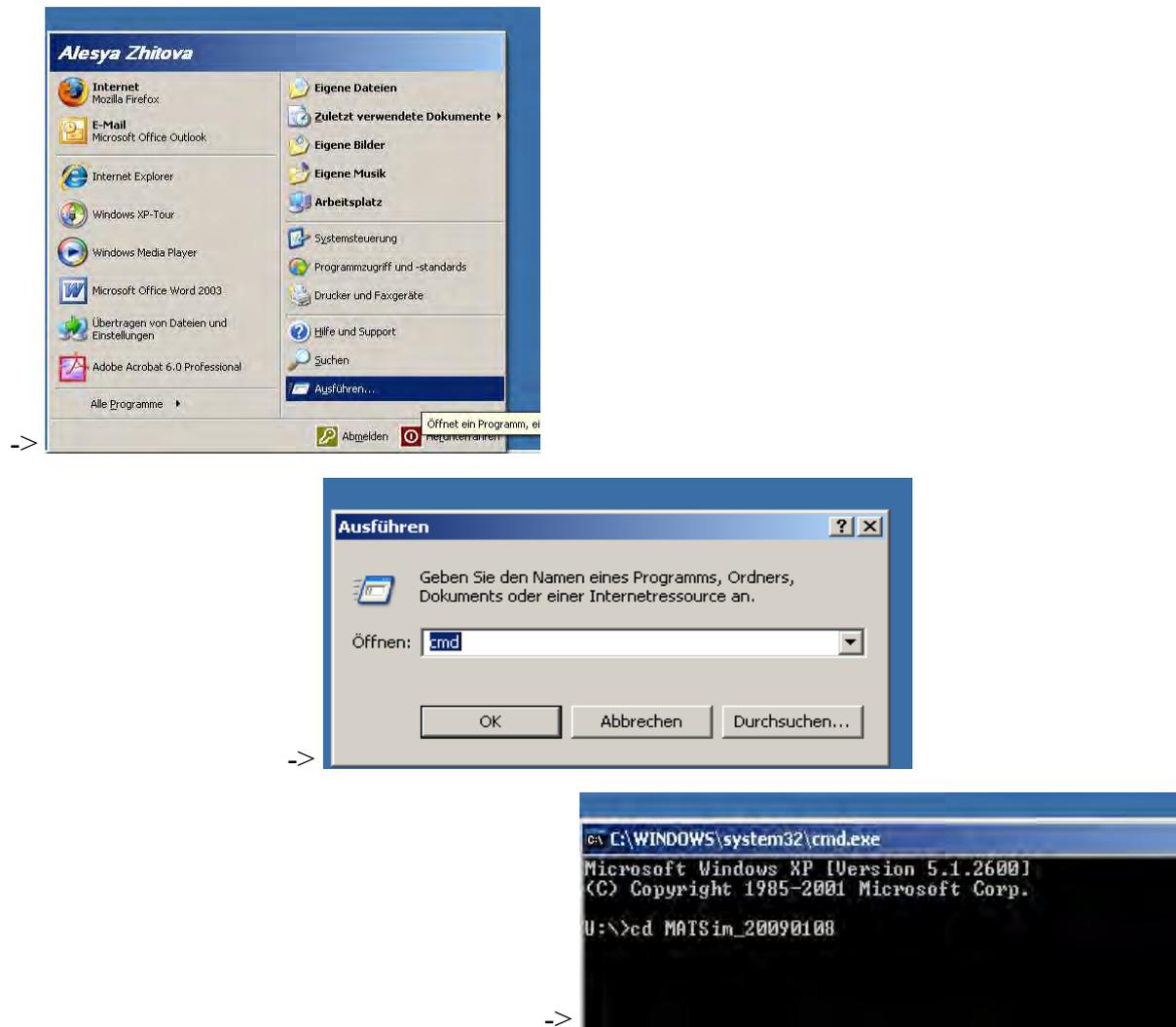


Abbildung 27: Darstellung der notwendigen Schritte, um den Verkehrssimulator MATSim aufzurufen.

3.2 Einarbeiten und Test des Simulators anhand eines Beispiels

Die erste Möglichkeit, die Software zu testen, wird dem Nutzer durch ein bereit gestelltes einfaches Beispiel gegeben. In diesem Beispiel wurden 100 Agenten generiert, die jeder zu seiner bestimmten Zeit das Haus in Richtung Arbeit verlässt, sich über eine bestimmte Route (von der Strecke „link 1“ zur Strecke „link 20“ über die Knoten 2, 7 und 12) mit einem bestimmten Verkehrsmittel („car“) fortbewegt, abhängig von der eigenen Arbeitszeit in der Arbeit verbleibt und über eine andere Route (von der Strecke „link 20“ zur Strecke „link 1“

über die Knoten 13, 14 und 15) nach Hause zurückkehrt (vgl. Abbildung 29). In der in der Dokumentation verfügbaren pdf-Datei „Getting started with MATSim“ sind die Schritte, wie man sich das Beispiel anschauen kann, detailliert beschrieben:

- Aufruf des Verzeichnisses

```
U:\MATSim_20090108>java -cp MATSim_r4776.jar org.matsim.utils.vis.netvis.NetVis
```

- Wahl der interessierenden Datei, wobei das File-Format auf „alle Dateien“ umgestellt werden muss. Um dieses Musterbeispiel ablaufen zu lassen, ist aus dem Verzeichnis die Datei mit dem folgenden Weg zu wählen: output/ITERS/it.0/SnapshotCONFIG.vis

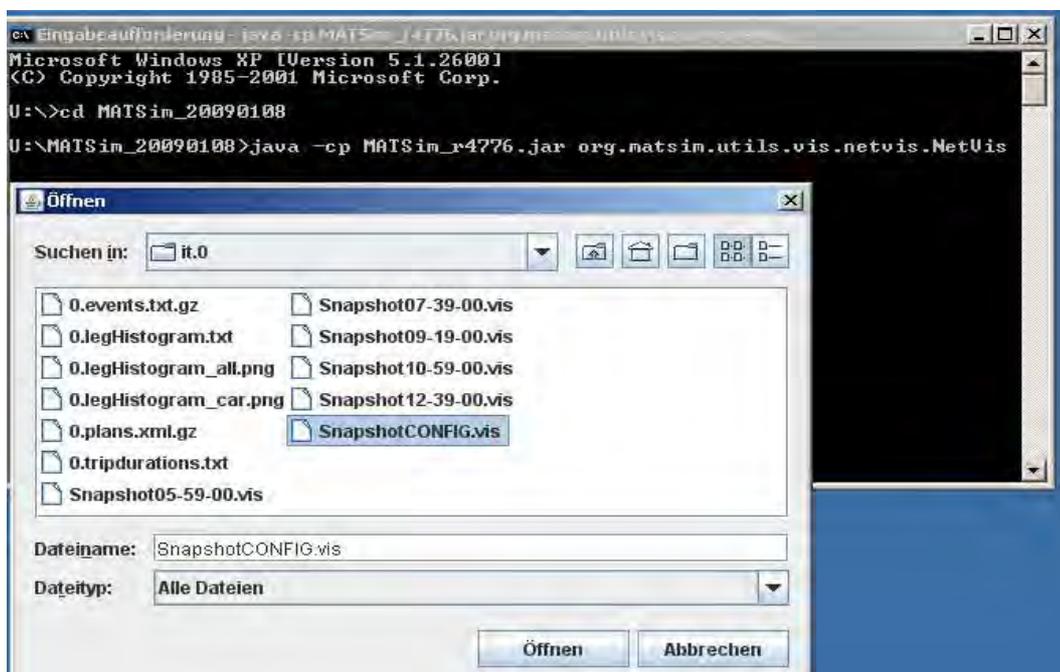


Abbildung 28: Aufruf des Beispiels von der Homepage des Verkehrssimulators MATSim.

Abbildung 29 zeigt dieses Beispiel. Das Netz besteht aus 23 Strecken und 15 Knoten, ihre Nummerierung kann dabei nach Belieben ein- und ausgeschaltet werden. In der oberen Leiste kann die Simulation angehalten, zurückgespult oder vorgezogen werden. In der Abbildung wurde die laufende Simulation in dem Moment festgehalten, in dem sich die ersten Agenten, die der Programmierung nach um 05:58:30, 05:59:00 und 05:59:30 starteten, bereits ein Stück Weg zurückgelegt haben. Die Simulation wurde für die Zeit von 05:59:00 bis 13:27:00 programmiert.

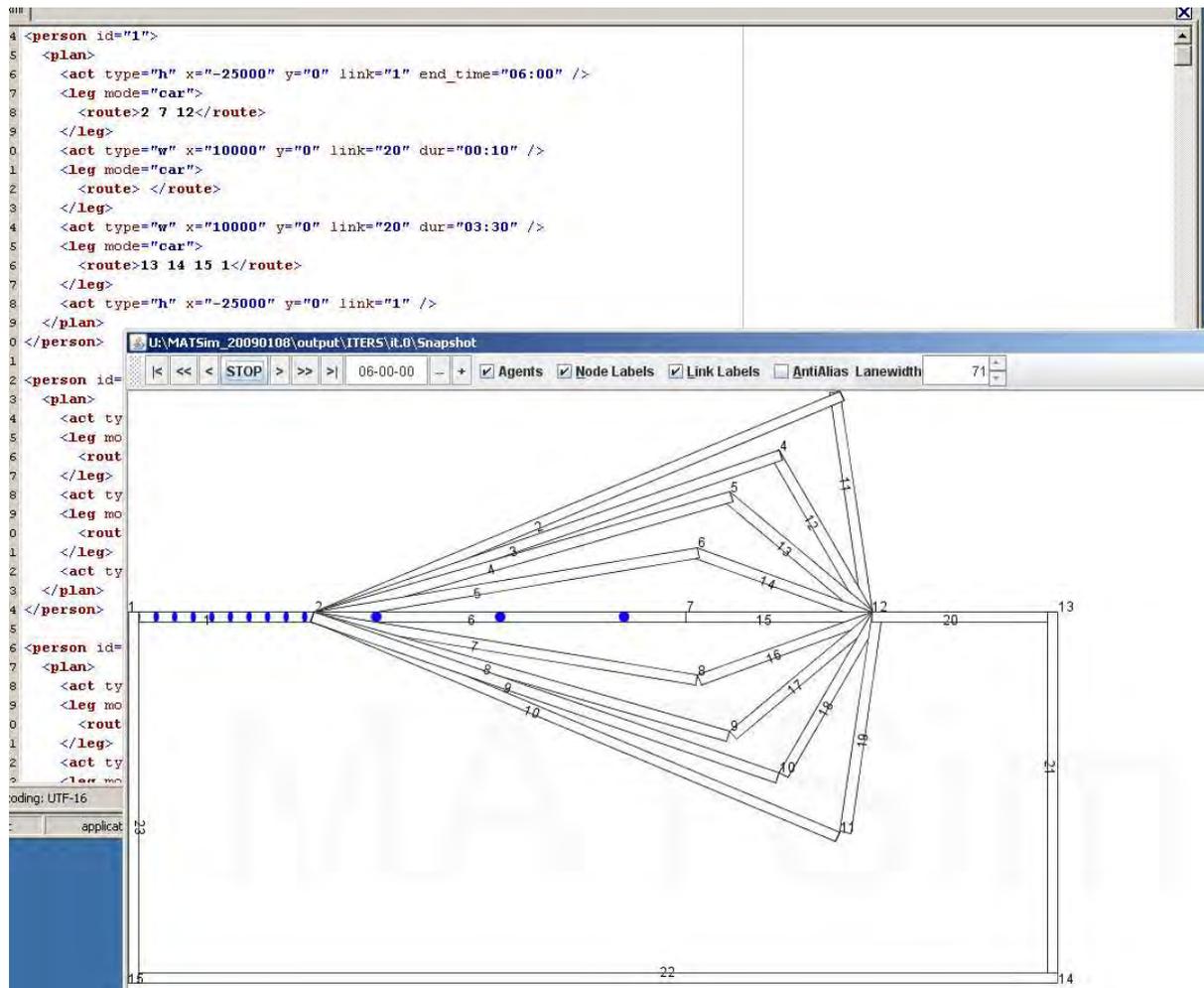


Abbildung 29: Netzwerk und Agenten aus dem Beispiel von der Homepage des Verkehrssimulators MATSim sowie eine XML-Datei zur Beschreibung dieser Agenten: z.B. die Person mit der Identifikationsnummer „id = 1“ verlässt das Haus „h“ mit den Koordinaten (-25000;0) auf der Strecke „link = 1“ um 06:00, fährt 10 Minuten „dur = 00:10“ mit dem Auto „car“ über die Knoten 2, 7 und 12 zur Arbeit „w“ mit den Koordinaten (10000;0) auf der Strecke „link = 20“, verbleibt in der Arbeit 3 Stunden und 30 Minuten und fährt zurück über die Knoten 13, 14, und 15 nach Hause „h“.

In diesem auf der Homepage des Verkehrssimulators MATSim angeführten Beispiel wird zwischen Singleiteration und Multipleiteration unterschieden. Bei der Singleiteration läuft die Simulation nur einmal, und zwar so wie sie ursprünglich programmiert wurde, ab: Jeder Agent führt einmal seinen initialen Plan durch (s. Abbildung 29). In dem Beispiel sieht man nach dem Ablauf der Singleiteration, dass bei dieser Wahl der Aktionspläne der Agenten sowohl auf der Hinfahrt als auch auf der Rückfahrt zu den Peak-Zeiten mit der maximalen Nachfrage zum stockenden Verkehr und der Staubildung kommt, die sich nur langsam auflösen.

Die vorhandene Situation kann durch die Erhöhung der Anzahl der Iterationen verbessert werden. Eine Simulation, die so programmiert wurde, dass sie mehrmals in Folge von vorne

abläuft, wird in diesem Beispiel als MultipleIteration¹⁶ bezeichnet. Dabei werden bei jeder neuen Iteration die initialen Pläne der Agenten in der abgeänderten, optimierten Form durchgeführt. Zur Optimierung der Agentenpläne werden in diesem Beispiel folgende zwei Module verwendet:

- „ReRoute“-Modul: Wahl neuer Routen/Wege und
- „TimeAllocationMutator“-Modul: Wahl neuer Zeiten für die Aktivitäten der Agenten

In diesem Beispiel ändert sich die Stausituation bei der Optimierung mit dem „ReRoute“-Modul nicht wesentlich auch beim Ablauf der zehnten Iteration, obwohl die Agenten jede neue Simulation andere, optimierte Routen wählen, die Routen, die für sie persönlich den meisten Nutzen (hier: kürzere Reisezeit durch kürzere Wege oder die Vermeidung des Staus durch die Ausweichung auf andere Wege) versprechen, da das Netz keine weiteren Optimierungsmöglichkeiten hinsichtlich der Wege zulässt (Zulauf aller Wege auf einen Knoten).

Bei der Optimierung mit dem „TimeAllocationMutator“-Modul wird die Abfahrtszeit der Agenten nach dem Zufallsprinzip geändert. Auf solche Weise fahren z.B. im Verlauf der neunzehnten Iteration die Agenten in den annähernd gleichen Zeitabständen gleichmäßig über 60 Sekunden verteilt los, wodurch es fast zu keiner Staubildung auf der Stammstrecke kommt und keiner auf die längeren Wege ausweichen muss. Dieses Szenario weicht aber eher von der Realität ab, denn die Situation der Straßenüberlastung in der Peak-Zeit wird in diesem Fall nicht simuliert. Sie wird durch gleichmäßig verteilte Abfahrtszeiten vermieden. Die Abbildungen 30 und 31 zeigen die neunte und die neunzehnte Iteration der MultipleIteration-Simulation, bei der die Optimierung der Agentenpläne nach dem „ReRoute“-Modul programmiert wurde. In der Abbildung 32 dagegen wurde die Optimierung der initialen Pläne der Agenten nach dem „TimeAllocationMutator“-Modul durchgeführt. Beide Optimierungsstrategien hätten auch gleichzeitig angewendet werden können. In dem Beispiel wurde diese Variante aber nicht in Betracht gezogen.

¹⁶ Die Simulationen in Form von MultipleIteration sind die Regel, da sie zu den besseren, realitätsnäheren Ergebnissen führen. Das Wort „MultipleIteration“ wurde daher nur in diesem Beispiel verwendet, um den Unterschied zwischen der einfachen einmaligen Simulation und der Simulation mit den integrierten Optimierungsmodulen zu verdeutlichen.

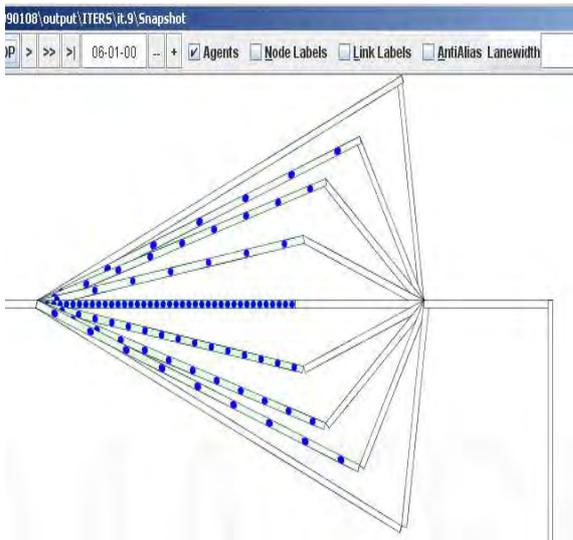


Abbildung 30: Ablauf der 9. Iteration

Beim Ablauf der 9. Iteration wählt etwa die Hälfte der Agenten alternative Routen, um das Stehen im Stau auf der geraden Strecke zu vermeiden, bei der 19. Iteration entscheiden sich wesentlich mehr Agenten für andere Wege. Die Zeit der Stauauflösung reduziert sich (bei der 19. Iteration fast um die Hälfte).

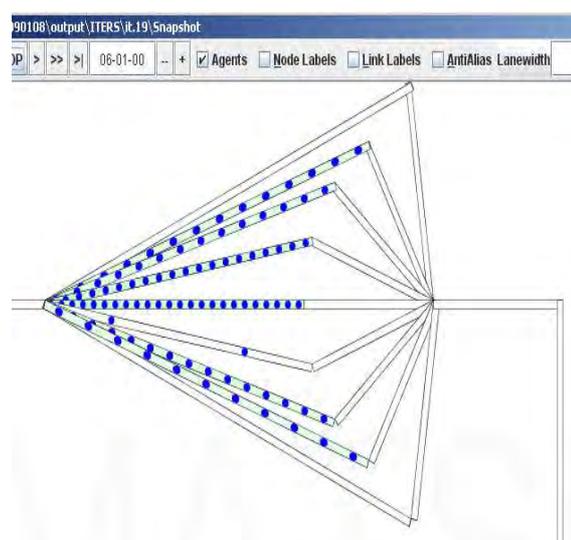
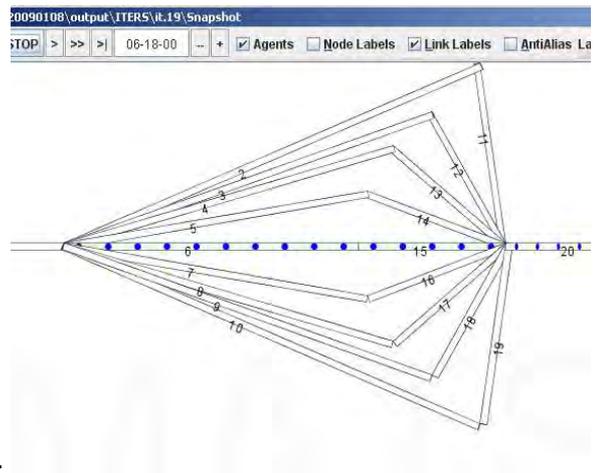
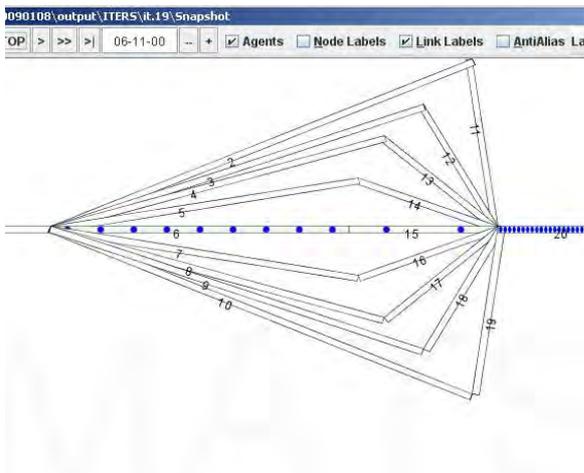


Abbildung 31: Ablauf der 19. Iteration



->

Abbildung 32: Ablauf der 19. Iteration bei der Optimierung der Simulation mit dem Modul „TimeAllocationMutator“. Durch gleichmäßig verteilte Abfahrtszeiten der Agenten kommt es fast zu keiner Staubildung auf der geraden Stammstrecke

Durch Änderungen in der Konfigurationsdateien configs/singleiteration.xml bzw. configs/multipleiterations.xml im Text-Editor und erneutes Ablaufen der Simulation, sowie die erneute Visualisierung kann man die Simulationen beliebig modifizieren und erneut abspielen lassen, um Änderungen auch optisch erfassen zu können und die Richtigkeit der Korrekturen ggf. kontrollieren zu können. In den Dateien network.xml und plans100.xml kann man zudem die Geometrie des Netzwerkes, die Anzahl und die Pläne einzelner Agenten ändern.

4 Kopplungsmöglichkeiten von MATSim und SinoSim

Wie bereits im Kapitel 1 erwähnt wurde, ist das primäre Ziel des Verkehrssimulators MATSim das individualisierte Modellieren jedes einzelnen Agenten und seiner Tagesabläufe. Aus der Summe aller Agenten kristallisiert sich dabei ein Verkehrsaufkommen bzw. ein Verkehrszustand heraus. Der Personenstromsimulator SinoSim, der im Rahmen des Forschungsvorhabens REPKA in Zusammenarbeit zwischen der TU München und Siemens entwickelt wird, hat dagegen zum Ziel, die Personenströme unter bestimmten Bedingungen, d.h. in einer definierten Situation, so realitätsnah wie möglich darzustellen (auch aus der psychologischen Sicht: überholen, drängeln, nah vorbei gehen – gehören auch dazu, was mit MATSim nicht simulierbar ist). Dabei besteht jedes Szenario aus einem oder mehreren Levels (z.B. Stockwerke in einem Gebäude), in jedem Level wird dann die Geometrie (z.B. Räumlichkeiten, Ausgänge, Hindernisse, Orientierungspunkte) vorgegeben, und schließlich Quellen (Ausgangsorte) und Senken (Zielpunkte) für die Fußgänger festgelegt. Somit werden die Fußgänger nicht jeder einzeln individuell implementiert, sondern nach Mengenvorgaben von den Quellen erzeugt und nach den Kapazitäten von den Senken aufgenommen. Das wird auf Basis eines mikroskopischen Modells (zellulärer Automat auf der Basis eines Hexagonalgitters) und der Potenziale realisiert. Pro Zelle wird dabei ein Wert aus der Summe der drei Potenziale (das Anziehungspotenzial der Senken/Orientierungspunkte, das Abstoßungspotenzial von den Hindernissen und das Abstoßungspotenzial von den anderen Fußgängern) berechnet. Da die Fußgänger für den nächsten Zug immer die Zelle mit dem niedrigsten Wert aussuchen, bewegen sie sich entlang der Gradienten von ihren Quellen zu ihren Senken. Unterwegs interagieren sie miteinander.

Im Prinzip können die Unterschiede und Gemeinsamkeiten der beiden Simulatoren in einer Übersichtstabelle zusammen gefasst werden:

	Verkehrssimulator MATSim	Personensimulator SinoSim
	Unterschiede	
Zielsetzung	<ul style="list-style-type: none"> • das individualisierte Modellieren der Tagesabläufe jedes einzelnen Agenten, • Simulation des daraus entstehenden Verkehrsaufkommens 	Simulation von realitätsnahem Fußgängerverhalten in Echtzeit

Ansatz	<ul style="list-style-type: none"> • Genetischer mikroskopischer Algorithmus, • modularer Aufbau 	Mikroskopisch auf Basis von zellulären Automaten und Potenzialfeld-Modell
Eingangsdaten für die Simulationen	Netzwerke (Graphkanten, Knoten) und initiale Pläne (mit Events) der einzelnen Agenten	Topologie/Geometrie, Quellen, Senken, Hindernisse, Orientierungspunkte
Bewegungsmodell	<ul style="list-style-type: none"> • entlang der Kanten von einem Knoten zum anderen • Interaktion der Agenten und des daraus entstehenden Verkehrs ist nur durch die Anpassung der initialen Pläne der Agenten aneinander möglich 	<ul style="list-style-type: none"> • Von den Quellen zu den Senken • Interaktion der Personen untereinander (überholen, drängeln usw.) ist möglich
Optimierungsverfahren	kapazitätenabhängige (je nach Zeit und Kosten der Kante) Wahl der Routen, Abfahrtszeiten, Verkehrsmittel, Orte und der Aktivitätenketten zur Optimierung und der Abstimmung untereinander der initialen Pläne einzelner Agenten	Austausch des Zielpotenzials durch einen graph-basierten Ansatz mit Zwischenzielen und darauf Routenwahl nach dem euklidischen Abstand
	Gemeinsamkeiten	
	<ul style="list-style-type: none"> • Vorhandensein der Pläne (explizit/implizit) • Analyse der Fitness der Simulationen • jederzeitige Ausgabe der Zwischenresultate • Anwendung des Dijkstra-Algorithmus zum Finden der kürzesten Wege 	

Tabelle 2: Unterschiede und Gemeinsamkeiten der Simulatoren MATSim und SinoSim.

Wie aus der Tabelle 2 ersichtlich, grenzen neben der Zielsetzung zusätzlich auch die grundlegenden Unterschiede im konzeptionellen Aufbau die Kopplungsmöglichkeiten von MATSim und SinoSim ein. Und trotzdem wären gewisse Kopplungsmöglichkeiten denkbar. Theoretisch wäre es z.B. möglich, ähnliche Szenarien (identische Geometrie/Netzwerke/Nachfragen) sowohl in MATSim als auch in SinoSim zu generieren. Im ersten Schritt startet man dann mit der MATSim-Simulation, deren Ergebnisse (z.B. optimierte Pläne der Agenten, Kapazitäten der Routen, benötigte Zeiten usw.) im nächsten Schritt als Input für SinoSim übernommen werden. Nach dem Ablauf der SinoSim-Simulation werden deren Ergebnisse (z.B. neue Pläne für Agenten) wiederum in MATSim als Input übergeben. Zum einen könnte man dabei überprüfen, ob die Simulationen miteinander konvergieren, zum anderen wäre es möglich, die Optimierung mit SinoSim mit der eigenen Optimierung von MATSim zu vergleichen, um dadurch vielleicht die Unstimmigkeiten der SinoSim-Simulationen ausfindig zu machen oder daraus gewisse Ansätze zur Verbesserung der Simulationsergebnisse hinsichtlich der Realitätsnähe zu gewinnen.

Somit würde eine solche Kopplung der beiden Simulatoren eine Möglichkeit zum Abgleich zwischen den Ergebnissen von zwei unterschiedlichen Ansätzen bieten. Diese Vorgehensweise könnte neue Erkenntnisse oder zusätzliche Optimierungsmöglichkeiten liefern.

5 Zusammenfassung und Ausblick

Diese Arbeit untersucht, inwieweit der als OpenSource-Software verfügbare Verkehrssimulator MATSim mit dem im Forschungsvorhaben REPKA zu entwickelnden Personenstromsimulator gekoppelt werden kann.

Die Arbeit beginnt mit der allgemeinen Vorstellung des Verkehrssimulators MATSim. Dabei werden auch einige Simulations-Projekte genannt, in denen der Simulator bereits zur Anwendung gekommen ist. Diese Beispiele sind in späteren Kapiteln die Grundlage für die Untersuchungen von Laufzeit und Speicherplatz, die mit Hilfe der Literaturrecherche durchgeführt werden.

Mit dem Ziel, die Funktionsweise des MATSim aufzuzeigen, wird außerdem ausführlich auf den konzeptionellen Aufbau des Simulators eingegangen. Eine wichtige Rolle spielt dabei die detaillierte Beschreibung der Vorgehensweise beim Generieren eines neuen Szenario, sowohl theoretisch als auch praktisch anhand eines Beispiels von der Homepage des Verkehrssimulators.

Anschließend werden Kopplungsmöglichkeiten für gemeinsame Szenarien von MATSim und dem im Rahmen des Forschungsprojektes REPKA zu entwickelnden Personenstromsimulator SinoSim erarbeitet. Daraus folgt, dass aufgrund der Unterschiede in der Zielsetzung und dem konzeptionellen Aufbau der beiden Simulatoren solche Kopplungsmöglichkeiten nur bedingt gegeben sind. Dennoch wäre es interessant, im nächsten Schritt zu versuchen, diese Kopplungsmöglichkeiten genauer zu untersuchen und zu realisieren, um ihre Ergebnisse zu analysieren und daraus eventuell Aussagen zur Qualität der Simulationen zu machen oder neue Erkenntnisse bzw. Optimierungsmöglichkeiten zu gewinnen. Eine Realisierung dieser Kopplungsmöglichkeiten ist nicht Gegenstand dieser Arbeit und stellt bei Bedarf seitens REPKA-Projektentwickler einen weiteren möglichen Untersuchungsschwerpunkt dar.

Literaturverzeichnis:

[1] NEUMANN, A (2008) Modellierung und Evaluation von Lichtsignalanlagen in Queue-Simulationen. Diplomarbeit, Berlin, Germany.

<https://svn.vsp.tu-berlin.de/repos/public-svn/publications/vspwp/2008/08-24/Neumann2008Dipl.pdf>

[2] RIESER, M. (2009) Anwendungen von Multiagenten-Simulationen.

<http://www.vsp.tu-berlin.de/>

fileadmin/a0533/uploads/lehre/LV/SS09/L050/20090602_06_MATSimAnwendungHandout.pdf

[3] BALMER, M. and M. Rieser and K. Meister and D. Charypar and N. Lefebvre and K. Nagel and K.W. Axhausen (2008) MATSim-T: Architektur und Rechenzeiten. Paper presented at Heureka '08, Stuttgart, Germany, March 2008.

http://matsim.org/uploads/Heureka08_matsim-architecture.pdf

[4] MEISTER, K. and M. Rieser and F. Ciari and A. Horni and M. Balmer and K.W. Axhausen (2008) Anwendung eines agentenbasierten Modells der Verkehrsnachfrage auf die Schweiz. Paper presented at Heureka '08, Stuttgart, Germany, March 2008.

http://matsim.org/uploads/Heureka08_matsim-anwendung.pdf

[5] BALMER, M., K. Meister, M. Rieser, K. Nagel and K.W. Axhausen (2008) Agent-based simulation of travel demand: Structure and computational performance of MATSim-T, paper presented at the 2nd TRB Conference on Innovations in Travel Modeling, Portland, June 2008.

<http://matsim.org/uploads/BalmerEtAl2008ITM.pdf>

[6] BALMER, M.: Wie funktioniert eigentlich MATSim?

im Rahmen der Präsentationen: Axhausen, K.W. and Balmer, M. and Charypar, D. and Horni, A. (2007) Sience City 2007 presentations, Treffpunkt Science City, ETH Zurich, Switzerland.

http://matsim.org/uploads/ScienceCity2007_Balmer.pdf

[7] HORNI, A.: Vergleich Simulation - Zählungen. Wie gut ist unsere Simulation?

im Rahmen der Präsentationen: Axhausen, K.W. and Balmer, M. and Charypar, D. and Horni, A. (2007) Sience City 2007 presentations, Treffpunkt Science City, ETH Zurich, Switzerland.

http://matsim.org/uploads/ScienceCity2007_Horni.pdf

[8] CIARI, F., M. Balmer and K.W. Axhausen (2007) Mobility tool ownership and mode choice decision processes in multi-agent transportation simulation, Paper presented at the 6th Swiss Transport Research Conference, Ascona, Switzerland, September 2007.

<http://matsim.org/uploads/Ciari2007strc.pdf>

[9] SCHARF, R (2008) Die Gesetze des Verkehrschaos.

<http://www.pro-physik.de/Phy/leadArticle.do?laid=10422>

[10] MEISTER, K. (2004) Erzeugung kompletter Aktivitätenpläne für Haushalte mit genetischen Algorithmen. Diplomarbeit, IVT, ETH Zürich, Zurich, Switzerland.

<http://matsim.org/uploads/Meister2004master.pdf>

und

Die Dokumentation zum als OpenSource-Software verfügbaren Verkehrssimulator MATSim.

<http://www.matsim.org>

Erklärung

Ich versichere hiermit, dass ich die vorliegende Bachelorarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Datum

Unterschrift

Anhang A:

Analyse der für die Optimierung der Simulation „Verkehrsnachfrage der Schweiz“ benötigten Anzahl der Iterationen.

Aus Balmer [3]:

„6.2 Relaxation

Der relaxierte Zustand im MATSim-T ist dann erreicht, wenn durch Variation der Tagespläne der Nutzen für jeden Agent nicht mehr merklich verändert werden kann. Da bei jeder Iteration die schlechten Tagespläne nicht „überleben“, folgt daraus, dass beim relaxierten Zustand alle übrigen Pläne eines Agenten einen ähnlich großen Nutzen haben.

Bei der Simulation der Verkehrsnachfrage der Schweiz scheint sich nach Iteration 70 eine solche Kombination von Plänen zu ergeben, welche eine stabile Verkehrslage erzeugen, die robust gegen Veränderungen einzelner Agenten ist. Gute Pläne werden durch das Replanning verdoppelt und beibehalten, sofern sie auch gut sind. Dafür fallen schlechte Pläne weg, so dass zu letzt hauptsächlich gute Pläne übrig sind. Diese Effekte zeigen die Darstellungen in Abbildung 4. Die roten, resp. blauen Kurven zeigen für eine bestimmte Iteration die Anzahl Agenten auf, die zum gegebenen Zeitpunkt losfahren, resp. ankommen. Die grüne Kurve gibt Auskunft darüber, wie viele Agenten zurzeit im System unterwegs sind. In Iteration 15 erkennt man, dass um ca. 09:30 Uhr ein solcher „Breakdown“ stattfindet, und sich danach das System nicht mehr davon erholt (etwa 310'000 reisende Agenten um 20:00 Uhr). Die Situation in Iteration 70 ist nicht mehr so dramatisch, jedoch erkennt man einen kleineren Zusammenbruch während der Abendspitze. Nach dieser Iteration beginnt aber das System zu relaxieren, was sich im Histogramm widerspiegelt: Es bildet sich die zu erwartenden Morgen und Abendspitzen heraus, mit Rückgang des Verkehrsaufkommens während des Tages und kaum Verkehr in der Nacht (grüne Kurve von Iteration 100 aus Abbildung 4)“.

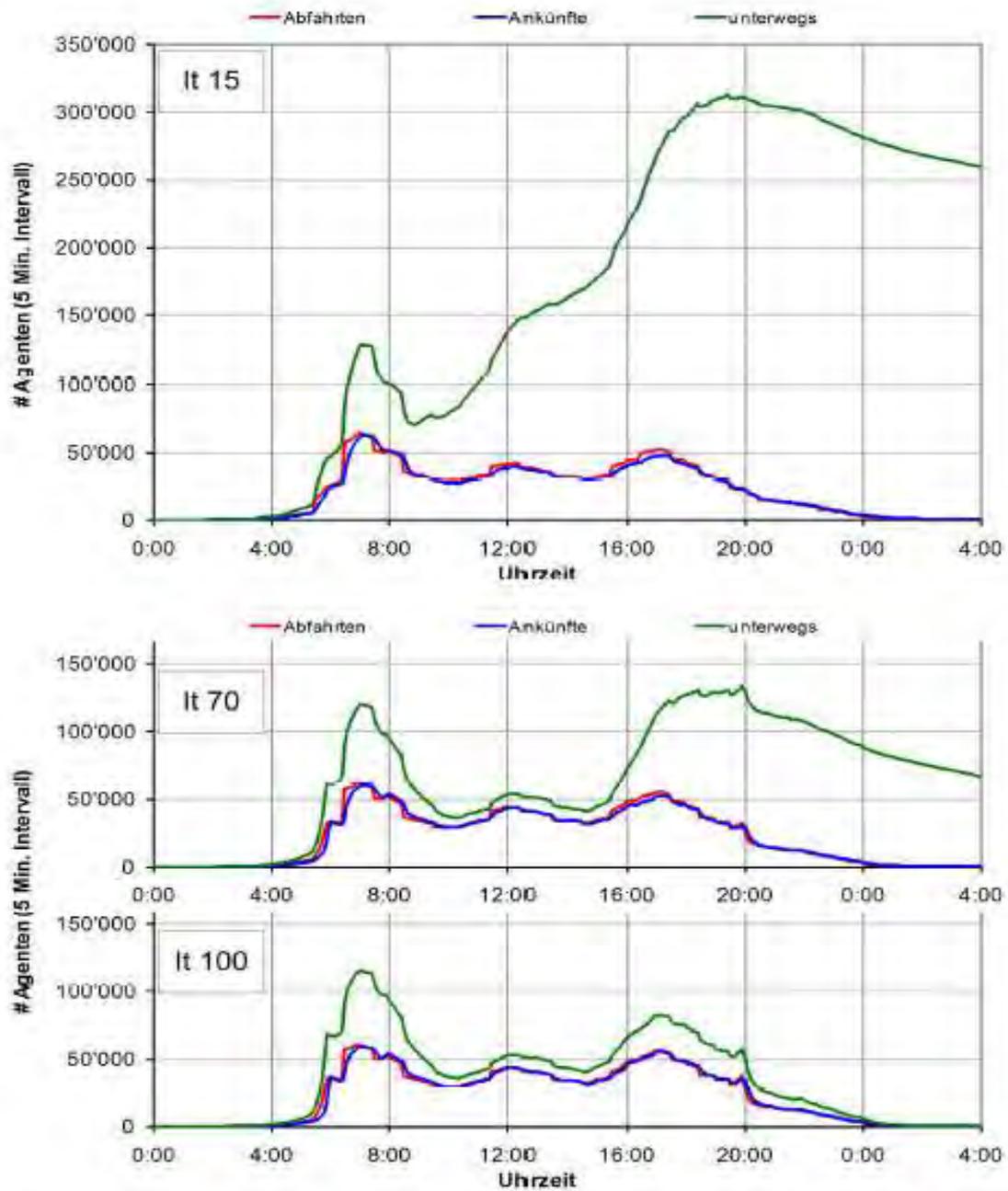


Abbildung 4: Abfahrts-, Ankunfts- und "unterwegs"-Histogramm für Iteration 15, 70 und 100

Anhang B: Publikationsverzeichnis

(Quelle: <http://www.matsim.org/publications>)

2008

Balmer, M. and M. Rieser and K. Meister and D. Charypar and N. Lefebvre and K. Nagel and K.W. Axhausen (2008) MATSim-T: Architektur und Rechenzeiten, paper presented at Heureka '08, Stuttgart, Germany, March 2008.

Balmer, M., K. Meister, M. Rieser, K. Nagel and K.W. Axhausen (2008) Agent-based simulation of travel demand: Structure and computational performance of MATSim-T, paper presented at the 2nd TRB Conference on Innovations in Travel Modeling, Portland, June 2008.

Meister, K. and M. Rieser and F. Ciari and A. Horni and M. Balmer and K.W. Axhausen (2008) Anwendung eines agentenbasierten Modells der Verkehrsnachfrage auf die Schweiz, paper presented at Heureka '08, Stuttgart, Germany, March 2008.

Rieser, M. U. Beuck, D. Grether, K. Nagel and K.W. Axhausen (2008) Multi-agent transport simulations and economic evaluation, Jahrbuch für Nationalökonomie und Statistik (forthcoming).

2007

Axhausen, K.W. and Balmer, M. and Charypar, D. and Horni, A. (2007) Science City 2007 presentations, Treffpunkt Science City, ETH Zurich, Switzerland.

Balmer, M. (2007) Travel demand modeling for multi-agent traffic simulations: Algorithms and systems, Dissertation, ETH Zurich, Switzerland.

Charypar, D., K. Nagel and K.W. Axhausen (2007) An event-driven queue-based microsimulation of traffic flow, Transportation Research Record, 2003, 35–40.

Ciari, F., M. Balmer and K.W. Axhausen (2007) Mobility tool ownership and mode choice decision processes in multi-agent transportation simulation, Paper presented at the 6th Swiss Transport Research Conference, Ascona, Switzerland, September 2007.

Grether, D. and Lämmel, G. and Rieser, M. and Strippgen, D. (2007) MATSim training 2007, Some presentations held during the visit of two colleagues from University of Pretoria, South Africa. VSP - TU Berlin, Dec. 2007.

Illenberger, J., G. Flötteröd and K. Nagel (2007) Enhancing MATSim with capabilities of within-day re-planning, paper presented at the IEEE Intelligent Transportation Systems Conference, Seattle, WA.

Lefebvre, N. and M. Balmer (2007) Fast shortest path computation in time-dependent traffic networks, paper presented at the 6th Swiss Transport Research Conference, Ascona, Switzerland, September 2007.

MATSim development team (ed.) (2007) MATSim: Aims, approach and implementation, IVT, ETH Zürich, Switzerland.

MATSim development team (ed.) (2007) Transport Systems' Planning and Transport Telematics at a Quick Glance, VSP, TU Berlin, Germany.

Rieser, M. and K. Nagel (2007) Network breakdown "at the edge of chaos" in multi-agent traffic simulations, *The European Physical Journal B*, doi 10.1140/epjb/e2008-00153-6.

Rieser, M. K. Nagel, U. Beuck, M. Balmer and J. Rügenapp (2007) Truly agent-oriented coupling of an activity-based demand generation with a multi-agent traffic simulation, *Transportation Research Record*, 2021, 10–17.

Rommel, C. (2007) Automatic feedback control applied to microscopically simulated traffic - the potential of route guidance in Berlin, VSP Working Paper, 07-16, TU Berlin, Berlin, Germany.

Titze, T. (2007) Entwicklung eines ÖV-Routingmoduls für Multiagentensimulationen, term project, Verkehrssystemplanung und Verkehrstelematik (VSP), TU Berlin, Berlin, Germany.

2006

Balmer, M., K. Nagel and B. Raney (2006) Agent-based demand modeling framework for large scale micro-simulations, *Transportation Research Record*, 1985, 125–134.

Charypar, D. and K. Nagel (2006) Q-learning for flexible learning of daily activity plans, *Transportation Research Record*, 1935, 163–169.

Charypar, D., K.W. Axhausen and K. Nagel (2006) Implementing activity-based models: Accelerating the replanning process of agents using an evolution strategy, paper presented at the 11th International Conference on Travel Behaviour Research, Kyoto, Japan, August 2006.

Hackney, J.K. and K.W. Axhausen (2006) An agent model of social network and travel behavior interdependence, paper presented at the 11th International Conference on Travel Behaviour Research, Kyoto, Japan, August 2006.

Meister, K. and Balmer, M. and Axhausen, K.W. and Nagel, K. (2006) planomat: A comprehensive scheduler for large-scale multi-agent transportation simulation, 6th Swiss Transport Research Conference, Monte Verità, Ascona, Switzerland.

Nagel, K. and F. Marchal (2006) Computational methods for multi-agent simulations of travel-behavior, in K.W. Axhausen (ed.) *Moving through nets: The physical and social dimensions of travel*, 131–188, Elsevier, Oxford.

Raney, B. and K. Nagel (2006) An improved framework for large-scale multi-agent simulations of travel behavior, in P. Rietveld, B. Jourquin and K. Westin (eds.) *Towards better performing European Transportation Systems*, 305–347, Routledge, London.

2005

Balmer, M. and Axhausen, K.W. and Nagel, K. (2005) An Agent Based Demand Modeling Framework for Large Scale Micro-Simulations, Working paper, 329, Institute for Transport Planning and Systems (IVT), ETH Zurich, Switzerland.

Balmer, M., B. Raney and K. Nagel (2005) Adjustment of activity timing and duration in an agent-based traffic flow simulation, in H.J.P. Timmermans (eds.): *Progress in Activity-based Analysis*, 91-114, Elsevier, Oxford.

Çetin, N. (2005) Large-scale parallel graph-based simulations, Dissertation, ETH Zurich, Switzerland.

Charypar, D. and Nagel, K. (2005) Generating complete all-day activity plans with genetic algorithms, *Transportation*, 32 (4) 369-397.

Marchal, F. and K. Nagel (2005) Computation of location choice of secondary activities in transportation models with cooperative agents, in F. Klügl, A. Bazzan and S. Ossowski (eds.) *Applications of Agent Technology in Traffic and Transportation*, 153–164, Birkhäuser, Basel, Switzerland.

Marchal, F. and K. Nagel (2005) Modeling location choice of secondary activities with a social network of cooperative agents, *Transportation Research Record*, 1935, 141–146.

Meister, K. and Balmer, M. and Axhausen, K.W. (2005) An improved replanning module for agent-based micro simulations of travel behavior, Working paper, 303, Institute for Transport Planning and Systems (IVT), ETH Zurich, Switzerland.

Meister, K., M. Frick and K.W. Axhausen (2005) A GA-based household scheduler, *Transportation*, 32 (5) 473–494.

Raney, B. (2005) Learning Framework for Large-Scale Multi-Agent Simulations, Dissertation, ETH Zurich, Switzerland.

2004

Frick, M. and Axhausen, K.W. (2004) Generating Synthetic Populations using IPF and Monte Carlo Techniques, 4th Swiss Transport Research Conference, Monte Verità, Ascona, Switzerland.

Meister, K. (2004) Erzeugung kompletter Aktivitätenpläne für Haushalte mit genetischen Algorithmen, Diploma theses, IVT, ETH Zurich, Zurich, Switzerland.

Nagel, K. (2004) Routing in iteration transportation simulations, in M. Schreckenberg and R. Selten (eds.) *Proceedings of the Workshop on Human Behaviour and Traffic Networks*, 305–318. Springer, Berlin, Germany.

Raney, B. and K. Nagel (2004) Iterative route planning for large-scale modular transportation simulations, *Future Generation Computer Systems*, 20 (7) 1101–1118.

2003

Çetin, N. and K. Nagel (2003) Parallel queue model approach to traffic micro-simulations, paper presented at the 82th Annual Meeting of the Transportation Research Board, Washington, D.C., January 2003.

Raney, B. and K. Nagel (2003) Truly agent-based strategy selection for transportation simulations, paper presented at the 82th Annual Meeting of the Transportation Research Board, Washington, D.C., 2003.

Raney, B., M. Balmer, K.W. Axhausen and K. Nagel (2003) Agent-based activities planning for an iterative traffic simulation of Switzerland, paper presented at the 10th International Conference on Travel Behavior Research, Lucerne, Switzerland, July 2003.

Raney, B., N. Çetin, A. Völlmy, M. Vrtic, K.W. Axhausen and K. Nagel (2003) An agent-based microsimulation model of Swiss travel: First results, *Networks and Spatial Economics*, 3 (1) 23–42.

Anhang C: Abbildungsverzeichnis

Abbildung 1: Homepage des Verkehrssimulators MATSim.....	6
Abbildung 2: Inhaltsverzeichnis der Dokumentation zum Verkehrssimulator MATSim.....	7
Abbildung 3: Simulation der gesamten schweizerischen Verkehrsnachfrage mit MATSim.....	8
Abbildung 4: Simulation der Durchquerung des Baregg-Tunnels in der Schweiz.....	9
Abbildung 5: Simulation der gesamten Verkehrsnachfrage Berlins.....	9
Abbildung 6: MATSim als modulares System.....	11
Abbildung 7: Der Prozess einer Simulation auf konzeptioneller Ebene.....	13
Abbildung 8: Beispiel für die Beschreibung eines Netzwerkes.....	14
Abbildung 9: Beispiel für die Beschreibung einer Person.	15
Abbildung 10: Events als zentrales Instrument der Simulation.	17
Abbildung 11: Zukünftige Event-Typen bei MATSim.	17
Abbildung 12: Für die Simulationen empfohlene Konfiguration der Module.....	18
Abbildung 13: Implementierung eines vierarmigen Knotenpunktes im MATSim.	21
Abbildung 14: Vergleich der Simulationsergebnisse mit den Daten der Zählstellen.....	23
Abbildung 15: Gründe für die Abweichungen.....	23
Abbildung 16: Vergleich Simulation - Zählstelle an der Autobahn A1, Baregg-Tunnel.....	25
Abbildung 17: Vergleich Simulation - Zählstelle an der Autobahn A3, Höhe Adliswil.....	25
Abbildung 18: Aufteilung der gesamten Rechenzeit des Simulationsprozesses.....	29
Abbildung 19: Rechenzeiten pro Prozessschritt.....	29
Abbildung 20: Rechenzeiten nach Modulen.....	30
Abbildung 21: Ergebnis für das Szenario „Student“.....	31
Abbildung 22: Ergebnis für das Szenario „Zwei Arbeiter mit einem Kind“.	31
Abbildung 23: Ergebnis für das Szenario „Kleinfamilie mit einem Schulkind“.....	32
Abbildung 24: Ergebnis für die Analyse der Rechenzeiten für alle drei Szenarien.....	33
Abbildung 25: Button „Download MATSim“.....	34
Abbildung 26: Download des MATSim von der Seite sourceforge.net.....	34
Abbildung 27: Aufruf des Verkehrssimulators MATSim.....	35
Abbildung 28: Aufruf des Beispiels von der Homepage des Verkehrssimulators MATSim...36	
Abbildung 29: Netzwerk, Agenten sowie eine XML-Datei aus dem Beispiel.....	37
Abbildung 30: Ablauf der 9. Iteration im Beispiel von der Homepage.....	39
Abbildung 31: Ablauf der 19. Iteration im Beispiel von der Homepage	39
Abbildung 32: Ablauf der 19. Iteration mit dem Modul „TimeAllocationMutator“.....	39

Anhang D: Tabellenverzeichnis

Tabelle 1: Übersicht der Event-Typen.....	16
Tabelle 2: Unterschiede und Gemeinsamkeiten der Simulatoren MATSim und SinoSim.....	41

Anhang E: Inhaltsverzeichnis der beigefügten CD

Bachelor-Arbeit als Word-Datei
Literatur als pdf-Dateien
Verwendeter Quellcode