Modelling Individuals for Simulating Contemporary Mobility Scenarios

by

JOHANNES NGUYEN

Thesis submitted in partial fulfilment of the requirements

of Edinburgh Napier University, for the award of

Doctor of Philosophy

School of Computing

Edinburgh Napier University

APRIL 2024

Author's Declaration

I hereby declare that the work presented in this thesis has not been submitted for any other degree or professional qualification, and that it is the result of my own independent work.

SIGNED: ....[redacted]............ DATE: ....

Abstract

Urban traffic is a system prone to overload, often approaching breakdown during rush hour times. Well-adjusted modifications of traffic policies, with appropriate interventions, promise potential improvements by inducing change in both individual as well as global system behaviour. However, truly effective measures are hard to identify, and testing in vivo is at least expensive and often hardly feasible. Agent-based traffic simulations are an established instrument to develop and assess policy interventions in silico but need to be further researched. In particular, better access to real-time information and a growing portfolio of mobility services have improved the flexibility in the personal mobility of individuals and thus require simulations to capture a more detailed modelling of the goals and purpose in their travel behaviour. Therefore, this thesis provides a systematic survey of existing traffic simulators, examining their ability to model individuals and their behaviour, and presents a modelling method based on semantic technology that allows preferences and personal objectives of individuals to be modelled as determining factors of agent decisions. The use of semantic technology helps to reduce the complexity during the modelling process. Furthermore, this thesis proposes a graphical notation that can capture the hierarchical structure of cause-effect relations in multi-agent models as well as a method to automatically extract cause-effect relations from the simulation at runtime. This allows implementations of simulation models to be reverse engineered, ensuring that the increasing complexity of the model does not further compromise the ability to understand the internal mechanisms of the simulation, which holds the potential to become an integral part of the structural validation of agent-based simulations. The thesis provides examples of implementation and gives demonstration of the proposed methods for artificial use cases.

Publications Associated with this Research

J. Nguyen, S. Powers, N. Urquhart *et al.*, 'Extending AGADE traffic to simulate auctions in shared mobility services', in *Proceedings of the 37th ECMS International Conference on Modelling and Simulation*, ECMS, 2023, pp. 201–207. DOI: 10.7148/2023-0201 - [Peer Reviewed]

J. Nguyen, S. Powers, N. Urquhart *et al.*, 'Multi-agent modelling notation (MAMN): A multi-layered graphical modelling notation for agent-based simulations', in *PRIMA 2022: Principles and Practice of Multi-Agent Systems*, Springer, 2022, pp. 640–649. DOI: 10.1007/978-3-031-21203-1_42 - [Peer Reviewed]

J. Nguyen, S. Powers, N. Urquhart *et al.*, 'Modelling the impact of individual preferences on traffic policies', *SN Computer Science*, vol. 3, no. 5, pp. 1–13, 2022. DOI: 10.1007/s42979-022-01253-3 - [Peer Reviewed]

J. Nguyen, S. Powers, N. Urquhart *et al.*, 'An overview of agent-based traffic simulators', *Transportation Research Interdisciplinary Perspectives*, vol. 12, 2021, ISSN: 2590-1982. DOI: 10.1016/j.trip.2021.100486 - [Peer Reviewed]

J. Nguyen, S. Powers, N. Urquhart *et al.*, 'Using AGADE traffic to analyse purpose-driven travel behaviour', in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, Springer, 2021, pp. 363–366. DOI: 10.1007/978-3-030-85739-4_33 - [Peer Reviewed]

J. Nguyen, S. Powers, N. Urquhart *et al.*, 'Modelling individual preferences to study and predict effects of traffic policies', in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, Springer, 2021, pp. 163–175. DOI: 10.1007/978-3-030-85739-4_14 - [Peer Reviewed]

J. Nguyen, S. Powers, N. Urquhart *et al.*, 'Using semantic technology to model persona for adaptable agents.', in *Proceedings of the 35th ECMS International Conference on Modelling and Simulation*, ECMS, 2021, pp. 172–178. DOI: 10.7148/2021-0172 - [Peer Reviewed]

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to my supervisors Neil Urquhart, Simon Powers and Michael Guckert for their unfailing support and guidance. I could not have undertaken this journey without their immense knowledge and expertise that have continuously presented me with new perspectives on my research. I must also thank my panel chair Emma Hart as well as the members of my thesis committee Peter Chapman and Gary Polhill for their insightful comments and encouragement. Acknowledgement is also due to Thomas Farrenkopf who has had a significant impact in preparing me for my academic work and has continued to support me throughout my Ph.D. studies. I would also like to thank my school teachers, fellow students and colleagues who have impacted and inspired me over the course of my educational and professional life. Lastly, I am deeply grateful to my family, especially my parents, as well as my friends and my love. Being with them has given me the necessary strength and composure. Their belief in me has kept my spirits and motivation high during this process. This endeavour would not have been possible without the generously provided funding from the Karl-Vossloh-Stiftung (S0047/10053/2019).

TABLE OF CONTENTS

List of Acronyms

ADAPTS	Agent-based Dynamic Activity Planning and Travel Scheduling
AOP	Aspect-oriented Programming
API	Application Programming Interface
ATL	Alternating-time Temporal Logic
ATS	Artificial Transportation System
BDI	Belief-Desire-Intention
BPMN	Business Process Model and Notation
CTL	Computation Tree Logic
EPL	Eclipse Public License
GPL	GNU General Public License
GUI	Graphical User Interface
ITS	Intelligent Transportation Systems
JADE	Java Agent Development Framework
LGPL	Lesser General Public License
LSP	Location-specific Probabilities
LT	Long-term
LTL	Linear Temporal Logic

MAMN Multi-Agent Modelling Notation

MT Mid-term

OD Origin-Destination

OSM OpenStreetMap

OWL Web Ontology Language

PDL Propositional Dynamic Logic

PhD Doctor of Philosophy

SCF Social Choice Functions

ST Short-term

SWRL Semantic Web Rule Language

TraSMAPI Traffic Simulation Manager Application Programming Interface

UML Unified Modeling Language

V&V Validation & Verification

W3C World Wide Web Consortium

XML Extensible Markup Language

List of Algorithms

LIST OF FIGURES

FIGURES		Page

List of Tables

Chapter 1

Introduction

A well-functioning transport system is an indispensable prerequisite for economic prosperity and development [14]. With the rapid increase in economic output and a growing world population, more people are drawn to urban centers. Common reasons for people relocating from rural areas to urban centers are better health care access, schools and career options [15]. This development is particularly noticeable within the younger generation, which induces rural exodus for the area. By 2050, 70% of the population are predicted to live in cities while only 30% remain in rural areas [16]. While the demand for living and office space in urban areas increases, available space and resources are limited [17]. Therefore, the majority of the urban population has moved into surrounding areas, and consequently causes higher traffic volume due to increasing commuter flows [18]. The need for individual mobility has grown to an unprecedented extent. Private road traffic such as daily commuting makes up a significant amount of the total traffic volume in urban areas (in 2016 commuting accounted for up to 74% of the total traffic) [16]. The repercussions for urban traffic are evident. Frequent traffic jams and the perpetual lack of parking space, are obvious indicators of a system in overload mode (infrastructure reaching a critical number of vehicles in the network) [19]. This shows that the current demand for flexible and individual mobility has exceeded the capacity limits of available infrastructure. In order to cope with these challenges, a fundamental change is required in the design of urban mobility.

Traffic simulation is an established means for studying and analysing current trends and patterns in mobility as well as their potential effects on future scenarios (e.g. [20], [21]). Global system behaviour is the result of the behaviour of a large set of individuals that perform actions following their personal objectives. The decision-making process of individuals is based on personal attitude as well as available world knowledge. New information that affects either of these aspects can lead to changes in individual behaviour. For example, the availability of smartphones as well as the increasing number of digital services both have led to changes in the behaviour of individuals in the context of personal mobility (see [22]). Information about routes, traffic jams or public transport connections can be accessed in real-time and thus has empowered individuals to determine their best travel option under almost complete information. While individuals used to carefully plan their trip prejourney, it has become remarkably easier to make spontaneous decisions en route. This has resulted in the increase of multimodal travel (travellers flexibly switching modes during their journey) that can be observed in urban traffic [23]. The growing supply of modern on-demand mobility services such as ride-hailing or sharing services targets precisely this type of short-term behaviour and will thus continue to intensify this development. With regard to research on traffic simulation, this signifies that the decision-making behaviour of individuals and their interactions with other individuals, objects and services in their surrounding environment is becoming more and more important and therefore needs to be considered when building simulation models. State-of-the-art research has been investigating traffic as an emergent phenomenon, rather than a problem that can be modelled from a global perspective where system behaviour is specified using aggregated and abstract parameters (e.g. [20], [21]). Higher-level system properties, such as the flow of traffic, emerge from the interactions of lower-level subsystems (e.g. individuals with personal objectives and autonomous behaviour) [24] which makes the application of multi-agent models a natural instrument.

As urban mobility is increasingly driven by personalised services, individuals are facing a growing number of travel options. This leads to individuals being more flexible in their personal mobility but also causes a growing complexity of decision-relevant information. Trying to appropriately reflect these changes in personal mobility places new requirements on traffic simulations. This research deals with the development of necessary concepts and methods to cope with the challenges involved in the implementation and application of individual-based traffic simulations. In particular, placing the individual and their behaviour in the center of attention requires elaborate techniques so that more details about personal objectives and decision behaviour can be modelled. By leveraging the ideas and concepts from semantic technology this type of information can be efficiently structured and the complexity of the modelling process can be reduced. Meanwhile, simulation models are becoming more complex and opaque due to the increased level of detail. Identifying and finding an appropriate representation for relevant cause-effect relations generates more insights into the internal mechanisms of the simulation at runtime.

1.1 Motivation

Traffic policies define the basic conditions under which individuals make decisions regarding their personal mobility. Changing them can have a significant impact on the behavioural patterns of individuals in traffic. Before designing and implementing new policies, the cause and effect of the current traffic situation must be scrutinised in order to develop measures that are accepted by the public and can eventually provide relief. However, identifying appropriate measures can be difficult as there is a wide range of policies that vary in their effectiveness as well as in the costs and time needed for their implementation. For example, raising awareness of the environmental effects of transportation choices can lead to changes in the short term. However, efforts to foster pro-environmental behaviour based on education and awareness have demonstrated the lowest success rate and thus can only lead to changes to a limited extent [25]. In contrast, the shift to new driving technologies such as electric vehicles can achieve more significant effects when it comes to reducing environmental pollution but is associated with considerable costs and can only be implemented in the long term. Creating a comprehensive charging infrastructure for electric vehicles is a large-scale project that involves a significant amount of capital investment and takes years to accomplish [26].

Public institutions and private companies are already working intensely on alternative strategies that exploit contemporary technological innovation [27], but need more elaborate tools for working out new mobility concepts that enable more flexibility in personal mobility and at the same time cope with the ongoing challenges of urban traffic. Available traffic simulations have focused on simulating traffic as the primary subject, thus not prioritising individuals pursuing personal objectives and the purpose of their journeys, such as travelling to work or going to shop for groceries. In order to achieve personal objectives, the movement of individuals to a different location should merely be regarded as a necessary means to an end. Consequently, road traffic itself should not be considered the sole focus when modelling traffic scenarios as individual traveller objectives are just as relevant. Thus, there is a need for traffic simulations to focus on the goals and purpose in the travel behaviour of individuals as this is crucial to appropriately reflect the current developments e.g. urban traffic is more and more driven by digital services that aim to facilitate real-time decisions in personal mobility.

1.2 Problem Statement

Placing the individual and their behaviour at the center of attention in traffic simulations creates additional complexity due to the modelled level of detail. It is therefore important to develop appropriate methods to facilitate the development of traffic simulations that are based on the actions of the individual. Building traffic simulations typically involves abstracting complex real-world processes into a reduced model that captures presumably the most important concepts and their relations for a given scenario, and implementing it as an executable piece of software. What is commonly referred to as the *modelling process* is a series of complex activities that require structured procedures (methods) as well as appropriate software tools. Simulation models are typically created for specific research objectives. For this purpose, information from various data sources needs to be collected and processed to formulate assumptions about the real world. This process of surveying data requires substantial effort and therefore simulations are often based on publicly accessible information sources. Assumptions are then transformed into a formal representation that can be implemented into an executable simulation software. To place the focus of traffic simulations on the individual, more details and thus more information about the individuals, their knowledge and their decision-making behaviour have to be incorporated into the simulation. This leads to new requirements in the activities of model development. For example, the additional complexity caused by the detailed modelling of individuals requires innovative modelling techniques. Furthermore, it is important to ensure that the increasing complexity of the model does not further compromise the ability to understand the internal mechanisms of the simulation.

The intention of this research is to develop and propose a set of methods that can be used to effectively build and work with individual-based traffic simulations. For this purpose, the current state of implementation in traffic simulation will be reviewed to evaluate and discuss gaps and limitations when it comes to simulating individuals and their behaviour in road traffic. Based on this, appropriate methods will be proposed to address relevant issues when placing the individual at the center of attention.

1.3 Aims

As early implementations of traffic simulations date back to the 1960s [28], a broad range of agent-based traffic simulators has been developed that each focus on different aspects of the transportation system. Therefore, traffic simulations also differ in the level of detail to which they consider individuals and their behaviour. Consequently, depending on the scope of application, different aspects of individual behaviour may have been included. The first aim of this research is to get an overview of the contemporary challenges in mobility as well as the wide spectrum of simulators and their scope of application. In this context, this research will specifically look at the role of individuals and the capabilities of simulators for modelling individuals and their behaviour. Based on the contemporary challenges in mobility, there will be new requirements for traffic simulations. Findings of the review on existing simulators will indicate gaps and limitations for which further research is required. The aim is then to develop appropriate concepts and methods to facilitate the modelling of individuals in traffic simulations. Furthermore, as the detailed modelling of individuals may lead to additional complexity in the simulation models, this thesis also aims to develop a method to generate more insights into the internal mechanisms of the simulation. Summarising this, the aims of this research include:

- Identify the limitations of existing traffic simulators in modelling individuals for contemporary challenges in mobility.
- Establish appropriate methodologies to improve the process of modelling individuals based on the identified requirements for traffic simulations.
- Develop a method to extract cause-effect relations in agent-based traffic simulations.

1.4 Research Questions

This thesis looks at the new requirements placed on traffic simulations that are caused by the ongoing developments in mobility e.g. individuals having better access to real-time information as well as the growing portfolio of mobility services. The resulting improvements in the flexibility of personal mobility thus require a more detailed view of individuals in traffic simulations. From the computer science perspective, a more detailed view of individuals and their behaviour increases the complexity of the simulation models. Thus there is a need for appropriate tools to efficiently handle the added complexity in the activities of the model development process. This thesis addresses the research gap to develop appropriate methods that can be implemented as software to facilitate the development of individual-based traffic simulations. To further specify the aims of this research the following research questions have been defined:

1. What are the main deficiencies in the modelling of individuals and their behaviour in existing agent-based traffic simulators?

2. How can the knowledge of individuals be modelled to capture their preferences and personal objectives as determining factors of decisions in mobility scenarios?
3. How can relevant cause-effect relations in agent-based traffic simulations be automatically extracted and formally represented?

1.5 Objectives

Based on the aims and research questions above, a number of research objectives have been determined in the scope of this thesis. In this section, the objectives of the research project will be briefly outlined while refraining from going into the details of the underlying methodology. The intention is to illustrate how the objectives have been structured and how they are linked. Details of the chosen research methodology for outlined objectives will be described in the relevant chapters of this thesis.

- Perform a systematic literature survey on available traffic simulators with regard to their area of application as well as implemented features for modelling individual behaviour and give a discussion on gaps and limitations that require further research.
- Develop a framework to efficiently model the preferences and knowledge of traveller agents, allowing the agents to be flexibly reused across different scenarios.
- Apply the modelling framework to build and simulate appropriate example use cases.
- Develop a framework to extract relevant information on cause-effect relations from agent-based simulations to generate a formalised representation that provides more insights into the internal mechanisms of the simulation.
- Apply the framework to extract relevant information on cause-effect relations from appropriate example simulations.

1.6 Summary of Contributions

The contributions of this research can be summarised as follows:

A systematic survey of available agent-based traffic simulators, published in the well-established journal *Transportation Research Interdisciplinary Perspectives*, that looks at the ability of simulators to model individuals and their behaviour based on contemporary areas of interest in mobility (see [4]). The synthesis of the broad range of information allowed key concepts in the field to be demonstrated in a coherent and organised structure. At the same time, the systematic survey provided an overview of the current state of implementation which is helpful for other researchers to find an appropriate simulation tool and highlight the lack of appropriate modelling concepts to capture the decision behaviour of individuals based on their preferences and personal objectives.

A modelling framework that is able to comprehensibly capture preferences and personal objectives as determining factors of individual decisions. In this approach, the implementation of agent knowledge has been separated from their operating behaviour (action selection). Using semantic technology, the knowledge of individuals can be structured in a form that ensures information can be easily managed. In particular, the framework reduces complexity in the modelling process by applying computer-based reasoning mechanisms. Furthermore, knowledge of individuals about their purpose of travel that may vary across different scenarios can be easily replaced and extended which improves the reusability of the implemented agents. Another aspect is that by modelling more details of the individual travellers using the proposed method, the effects of policies can be evaluated not only on global system behaviour but also on individuals, which is important when developing new traffic policies.

An extension of UML activity diagrams called Multi-Agent Modelling Notation (MAMN) that is able to model the multi-level property of cause-effect relations in agent-based systems.

A framework to automatically extract relevant information on *cause-effect relations* in agent-based simulations *and to represent* these relations in the proposed MAMN graph structure. This approach provides insight into the internal mechanisms of the simulation, allowing implementations of simulation models to be backwards/ reverse engineered. The systematic method helps to ensure completeness and correctness of the produced representations.

1.7 Outline of the Thesis

The remainder of the thesis is structured as follows:

Chapter 2: Background

This chapter provides an overview of the theoretical background of simulation theory, model development, game theory and agent-based methods which serve as important groundwork for this research.

Chapter 3: A Systematic Survey on Modelling Individuals Using Agent-based Traffic Simulators

To get an overview of the current state of implementation of existing traffic simulators to model individuals and their behaviour a systematic survey is given. This chapter also includes a discussion of challenges and limitations to be addressed in the scope of this research.

Chapter 4: Modelling Individual Preferences to Study and Predict Effects of Traffic Policies

This chapter proposes a structured method based on semantic technology for creating traffic simulations that are able to comprehensibly capture preferences and personal objectives as determining factors of individual decisions. The focus of this chapter is not to present a validated simulation model but to demonstrate how the proposed method can be used to model and simulate what-if scenarios when investigating the effects of policy interventions in simulations.

Chapter 5: Current State in Validation and Verification of Agent-based Simulations

As the detailed modelling of individuals increases the overall complexity of simulations, there is a need for appropriate methods to not further compromise the ability of researchers to understand them. For this purpose, this chapter reflects on current approaches for validation and verification and gives a discussion about the existing challenges when it comes to generating more insight into the internal mechanisms of individual-based simulations.

Chapter 6: A Graph-based Framework for Extracting Cause-Effect Relations from Agent-based Traffic Simulations

To extend agent-based simulations with more explanatory capabilities on their internal mechanisms, this chapter proposes a method for extracting relevant information on cause-effect relations of input and output variables from a simulation at runtime. Demonstration of the method is given by generating cause-effect graphs for different simulation models.

Chapter 7: Critical Evaluation

This chapter reflects on contributions to knowledge and discusses the benefits and limitations of the proposed methods as well as possible alternative research paths. Furthermore, a discussion is given of the informative value of the performed experiments as well as the current state of implementation.

Chapter 8: Conclusion and Future Research

Finally, this chapter draws final conclusions and gives suggestions for further research.

Chapter 2

Background

This chapter provides an overview of the theoretical concepts and notions that serve as important groundwork for this research. For this purpose, a description of the basic terminology used in the area of model development and simulations is given. More detailed coverage is given on the two subjects of *modelling* as well as *validation and verification*. With regard to the section on *modelling*, this work focuses on building models based on utility theory from economics. In particular, *game theory* covers important work on the abstraction and formalisation of real-world systems into quantifiable models. The analysis of these models provides insight into the behaviour of a system, which can be particularly useful for evaluating measures in policy-making. However, the question of *correctness* arises when policies are based on a reduced model of the real world. It is therefore important to *validate* and *verify* these models and simulations. This chapter gives a discussion of the different perspectives on simulation validation and verification, and briefly summarises applied techniques. Furthermore, as the focus of this research is on traffic simulation, an overview is given of the different types of simulation models. In particular, the role of multi-agent models is given special attention as there has been an increasing interest in their application for building traffic simulations.

2.1 Fundamentals on Systems, Models and Simulations

There is a wide range of literature that covers the theoretical foundations of simulation theory discussing the basic notions from different perspectives [29]–[31]. Although there can be minor differences in the terminology, the described concepts are in principle consistent. Issues in the real world are typically examined in a specific context which can also be described as a *system*. A *system* is a *spatially contained, logically coherent* and *temporally finite* entity that *comprises interdependent elements* [31]. For example, urban road traffic is a system that consists of many individuals as well as the available road infrastructure. It is *spatially contained* as available infrastructure is limited and thus can be considered a defined area. *Logical coherence* is an important requirement to determine whether something in the real world is considered part of the system which can be subjective depending on the perception of a specific human and/or a group of humans defining the system. In particular, there are many individuals that are located within the urban space, but only when they are moving within the road system, are they temporally taking on the role of a traffic participant which makes them part of the urban road traffic system. Individuals that are considered part of the traffic system interact with different elements of the infrastructure (e.g. traffic lights, traffic signs), as well as with each other. The decisions of one individual have an effect on the others which makes them *interdependent* e.g. at intersections or through congestion. *Temporal finiteness* is given when urban road traffic is viewed over a specified timeframe e.g. a day, a week or a year.

The complexity of these real-world systems makes it difficult, if not impossible, to describe them in their entirety [31]. *Models* are an *abstraction* of the real-world system. They capture the dynamics of a system by focusing on relevant aspects and their relations. Observations about the real world are recorded as *empirical data* which serves as a description of the system. The complexity of a given system determines the number of relevant aspects to be included in the model. Detailed models are likely to be complicated, hard to understand and difficult to work with [32]. Therefore, it is important to create models that are as simple as possible (see [33], [34]). Models should only be sufficiently complex to meet the modelling objectives [8]. Adding more aspects to the model may even reduce its accuracy, as at some point there is not enough data to support the expected level of complexity making it necessary to rely on assumptions (see Figure 2.1) (see [35] for a more comprehensive discussion on the complexity of models). Thus, it is important to ensure that necessary data on the real-world system is available and to acknowledge that models can only be as reliable as the data used for building them.

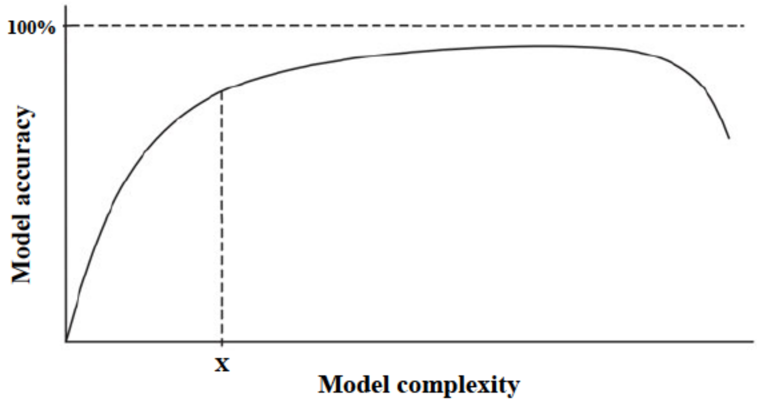

Figure 2.1: How model accuracy changes with increasing model complexity (see [8]).

As the number of aspects in the model increases, modelling and analytical methods may also become more complex. For example, relations between various aspects of a system can be mathematically represented, typically forming a system of difference or differential equations. Analysis of the model becomes more complex as the number of variables and equations as well as the amount of non-linearity increase. To handle this complexity, computer-based simulations are frequently used. Building computer-based simulations is an iterative process with different stages of model maturity (see Figure 2.2) [8]. In particular, information about the real-world system is gathered to create a *conceptual model*, which is then used to develop a *model design* (e.g. choosing algorithms and data structures) that can be implemented as a *computerised model*. [8] describes the *conceptual model* as a theoretical construct that exists within the mind of the modeller. It captures the abstracted understanding of the modeller about the real-world system. The conceptual model is defined by *objectives, model contents, inputs, outputs* as well as *assumptions* and *simplifications* [8]. *Objectives* describe the purpose of the model, e.g. a specific research problem to be investigated. These objectives have a significant impact on the *model content.* In particular, model content contains a set of relevant aspects from the real-world system and their relations. Thereby, the objectives specify the scope as well as the necessary level of detail for the model content. For example, the same traffic system can be viewed from different perspectives. Transportation planners typically look at the traffic system from a global perspective and therefore focus on different aspects related to social benefit rather than a mobility service provider that has a more customer-centric view and is looking to maximise profits. Since the objectives of both are fundamentally different, model content probably focuses on other aspects of the system. *Inputs* are variables of the research problem that when altered may lead to changing effects on the overall behaviour of the system. These resulting effects are captured and analysed in the *output* of the model. *Assumptions* are made when there are uncertainties about how the real world should be modelled whereas *simplifications* are deliberate abstractions to improve the manageability and transparency of the model.

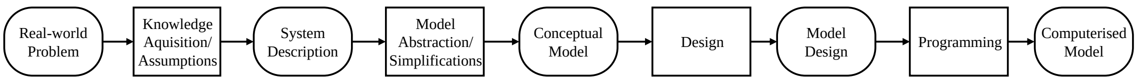

Figure 2.2: Model maturity (see [8]).

Creating the conceptual model is generally agreed to be the most difficult and also the most important activity in model development. While the creation of the conceptual model is described as a purely cognitive process that continuously takes place even beyond the completion of a simulation study, it may or may not be formally expressed [8]. For example, there have been different approaches for documenting the conceptual model such as listing assumptions and simplifications or creating variations of Petri nets, activity, logic flow or other types of process flow diagrams. Any type of documentation is considered an explicit representation of the conceptual model. The *computerised model* is a specific explicitation of the conceptual model that implements theories and assumptions from the conceptual model into an executable piece of code.

Ideally, the simulation model demonstrates system behaviour analogous to the observed real-world system. Based on this, it is typically assumed that the model is an appropriate representation of the system so that it can be used to make predictions about future behaviour [31]. Establishing trust and credibility in the output of simulations requires *validation* and *verification (V&V)* of simulation models [36]–[38]. Despite some occasional confusion or divergent uses of these notions, the relation between *validation* and *verification* has been argued to be distinct [39]. A commonly referenced definition of both is given by Law and Kelton [30], [40]:

- ***verification*** determines whether a conceptual model has been correctly transferred to a computerised model, whereas
- ***validation*** focuses on whether the conceptual and computerised model is an accurate representation of the real-world problem.

Validation and *verification (V&V)* of the simulation lead to recalibration or redesign of the model which makes model development a continuous activity. This concludes the basic overview of the terminology used in the area of model development and simulations. The following sections give a more detailed discussion on the two subjects *modelling* as well as *validation and verification* as these are particularly relevant for further research in this thesis.

2.2 Modelling

The abstraction of a given real-world system into a simulation model requires formalisation. In this context, a particular challenge lies in the representation of social behaviour and interaction. There are different views on how to formalise these types of concepts that have evolved from different disciplines e.g. psychology or economics. Each of these approaches has its strengths and limitations, but none of them is considered better than the others. Rather, it should be noted that theory is often not specified in such depth that there is only one way to implement or formalise it in a model. Sometimes the combination of ideas from different theories can help to overcome the limitations of one theory on its own. This research takes the utilitarian perspective from economics to model the behaviour and interactions of individuals in social systems. This approach relies extensively on mathematics to describe the behaviour and relations of a system which facilitates the implementation of the computerised model and comes with a large body of related work on model analysis and evaluation.

2.2.1 Studying Social Behaviour Using Games

Game theory is considered a subcategory of applied mathematics that is applied in economics and computer science to study strategic interactions of individuals. The term originally comes from strategy-based board games such as chess and was first used in a scientific context in 1944 [41]. Since then, a growing number of scientists have been working on this subject, discussing the problems of game theory from different perspectives [42], [43]. The research field is committed to studying decision-making and modelling of conflict situations in a mathematically quantified form. This enables formal analysis of decision behaviour and allows to determine the optimal course of action. In daily situations, when at least two individuals interact, conflict situations arise, both knowingly or subconsciously. Especially in an environment with shared limited resources, interaction between individuals often leads to competitive behaviour. Individuals observe their surrounding environment and when acting rationally, aim to achieve the best possible outcomes for themselves according to their personal objectives.

The same type of conflict situations can be observed in everyday mobility. Self-interested individuals such as travellers interact in an environment with limited and shared resources (infrastructure) and try to reach their destination through a supposedly optimal course of action. The decisions of one traveller ultimately have an effect on others. For example, when large groups of individuals, during rush hour, simultaneously choose to travel in private vehicles, this can lead to prolonged travel times due to traffic jams. This is why the application of game theoretic concepts is appropriate to create and analyse formal models of the traffic system, even though the vast number of travellers makes interaction complex and thus leads to numerous outcome scenarios.

2.2.2 Assumptions

Abstraction and formalisation of real-world scenarios as *games* requires assumptions. Classic game theory assumes individuals to behave completely rationally following the model of *Homo economicus* (see [44]). It is assumed that individuals always choose the most beneficial option when faced with multiple alternatives. A player in a game always has at least two possible actions. Each combination of selected actions by the players leads to a clear outcome. Outcomes are linked to utility functions that determine the personal utility (payoff) of players (see [45] for a formal definition). Players aim at maximising their personal utility while knowing the rules of the game as well as all possible payoffs. Usually, the decisions of one player influence the outcome of other players. It is therefore necessary for a player to anticipate the choices of other players when making a decision, assuming that every player knows about the set of possible options. These assumptions come with limitations in the modelling. It has been argued that in reality, complete information is rarely the case [46]. Furthermore, individuals in the real world often demonstrate behaviour that cannot be explained with rationality. The ultimatum game for example describes a scenario in which two players receive a certain payoff (e.g. 10 units) if both players accept [47]. However, the rules of the game state that a player A is responsible for splitting the amount and player B can then either accept or reject his share. In case of rejection, both players have to return the total payoff. In the event of a significantly unbalanced split, for example, 9 shares for player A and 1 share for player B, assuming completely rational behaviour, both players should accept the offer. Rationally, 1 share for player B is still better than no share. However, similar experiments from the real world have shown that in this situation, most of the participants in the position of player B will reject the offer as they consider the split to be unfair [48]. Concepts such as *bounded rationality* [49], *evolutionary game theory* [50] as well as non-deterministic modelling using probability distributions [51], [52] can help to overcome some of these limitations.

2.2.3 Network Games

Network games are a specific subcategory of classic game theory which are particularly suitable to capture the characteristics of road traffic as system behaviour is linked to the provided road network. Traffic participants can be considered players in a game who share the same resources in a network such as roads, infrastructure and mobility services. Interaction in such networks inevitably leads to conflict situations which can be modelled as games. Literature differentiates two types of network games [53]. The first type deals with games that are performed on already *formed networks*. Route selection on an existing road network is an example from everyday mobility. In addition to this, the second type is referred to as *network formation games*. This type examines scenarios in which a new network is formed through the establishment of links between various vertices. For example, given a scenario in which a number of individuals is travelling on the existing road infrastructure, there may be games that look at the formation of a new social network. Issuing a traffic warning and observing how information spreads is an example of this type of game (e.g. see [54] for a discussion). Sharing the information about the traffic warning creates a link between participants in the game and thus produces a social network that demonstrates the flow of information. This thesis focuses on games performed on formed networks. The following sections illustrate examples that are associated with this type of game.

A particular form of network games is *congestion games* [55], which are non-cooperative games in which players refrain from all communication. The payout of each player depends on the selected resources and the number of other players that also selected the same resource. The resources in these scenarios are shared and accessible to all players. One of the probably best-known examples is the Braess paradox [56]: In places where high traffic volume often leads to traffic jams and the demand for journeys is constant for the considered period of time, it can be assumed that infrastructure has reached full capacity. Attempts to solve this situation often include the construction of even larger roads or alternative routes. However, this approach causes redistribution of the overall traffic flow which can lead to even worse outcomes. Formally, the Braess paradox can be presented as a graph.

Let graph G be an ordered set of $\{V, E, F_G\}$ in which V is a nonempty set of vertices, E is a set of edges, and F_G is an incidence function that assigns an unordered pair of vertices to each edge, i.e. $E \subseteq V \times V$. Based on this, let $V = \{A, B, C, D\}$ and E be the set of ordered tuples $E = \{e_1, e_2, e_3, e_4\}$ with $e_1 = (A, B)$, $e_2 = (C, D)$, $e_3 = (B, D)$ and $e_4 = (A, C)$. Furthermore, the costs for travelling over the edges are defined $c(e_1) = c(e_2) = 1$, $c(e_3) = n(e_3)$ and $c(e_4) = n(e_4)$ with $n(e)$ indicating the amount of players travelling on that particular edge. All players start at vertex A trying to reach the target destination vertex D. In doing so, a player has to travel through a route R which in this game is an n-tuple of edges e.g. (e_1, e_2) to which applies $\pi_2(e_1) = \pi_1(e_2)$ with $\pi_i(e)$ being the i-th item in a tuple $e \in E$. In particular, players can choose between two options, either to move along the upper route (e_1, e_3) or to use the lower route (e_4, e_2) (see Figure 2.3).

In a second iteration, assume that an edge $e_5 = (B, C)$ is added to the graph with $c(e_5) = 0$. This changes the situation and it is now possible to change directions when arriving at vertex B. With the addition of e_5, costs in the system have changed and the most efficient route appears to be traversing over the edges e_1, e_5 and e_2. Considering this fact, all players are now choosing the new allegedly fastest route, resulting in all traffic being concentrated on the same edges, causing costs to be maximised. This ultimately leads to a worse global outcome [57].

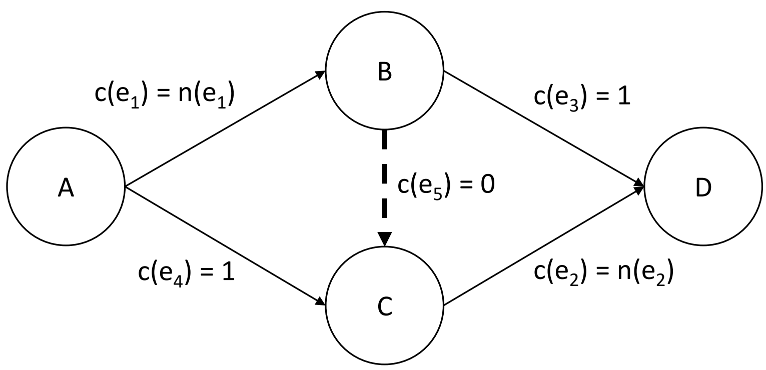

Figure 2.3: Graph representation of the Braess paradox.

Other forms of network games include *load balancing* and *cover games*. Load balance games are also described as scheduling problems on parallel different-capacity channels [53]. In these scenarios, there are a number of jobs that are to be distributed among a set of different processors. Each job can have a different job volume and processors may be working at different speed levels. The final *load* of a processor is defined by the sum of all job volumes executed on that particular processor. An important key indicator is the *delay* which basically specifies the required processing time of a job on that particular processor. This indicator is defined as the ratio of the load of a processor and its processing speed. In the context of mobility, the application of such games can be found in the planning of public transport schedules as well as speed planning for road traffic. The primary objective of these games is to minimise social costs of the overall system while players (travellers) selfishly try to minimise their own costs such as the required travel time, by choosing their best route option. Social costs are the aggregation of adverse effects that result from the strategic interactions and decisions of individual players within a game.

The KP-Model is an example of a load balancing game that is often referred to in the literature (see [58]). Assuming a graph that consists of the two vertices $V = \{A, B\}$ and E a set of edges for which applies $\forall e \in E, e = (A, B)$ (see Figure 2.4). Based on this, the graph includes m parallel routes from vertex A to vertex B, with m being the number of elements in set E. Each $e \in E$ is assigned a cost $c(e)$. All players are choosing a route to move from vertex A to vertex B. The KP-model is a game with mixed strategies meaning that each route option is assigned a probability with which it is selected. The costs of a route $c(e_n)$ are again dependent on the number of players who have chosen the same route. Social costs are defined as $\sum_{n=1}^{m} c(e_n)$ for which applies $e_n \in E$. Similar to the first iteration of the Braess paradox, once a player starts moving there are no options for changing routes. This is denoted as *unsplittable traffic*. The purpose of this game is to minimise the social costs. The KP-model can also be regarded as a weighted congestion game.

Cover games are based on the same problem setting. Instead of minimisation of job delay, cover games aim at maximising job delay [53]. Although this concept initially

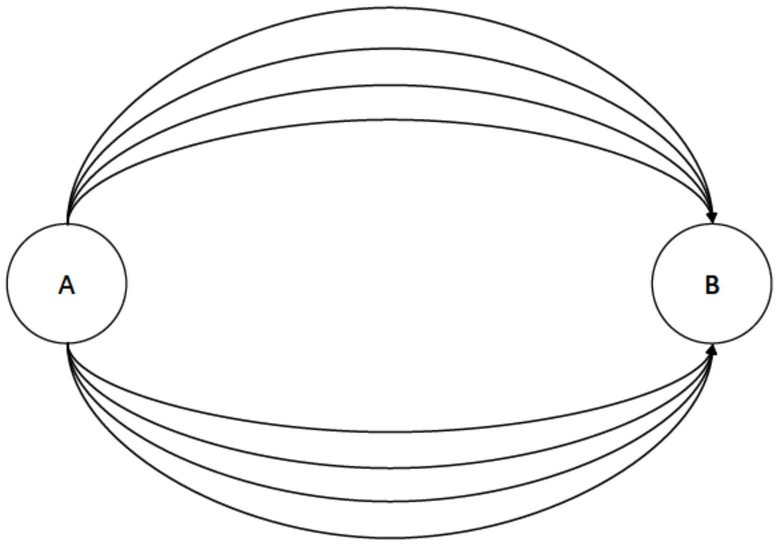

Figure 2.4: Example graph with m=8 parallel routes from A to B.

appears counterintuitive, there are scenarios in which it is more favourable to achieve processors in a system to be loaded at full capacity with minimum downtime. For example, if a service is paid by the hour and the revenues go to the benefit of the social community. In the context of mobility, this could be sightseeing buses for tourists that are managed by the city.

2.2.4 Equilibrium Situations

Given the diverse combinations of individual decisions, traffic flow forecasting at first sight can be difficult. However, it is well known that at certain times traffic jams keep occurring at the same locations. This is the result of individuals repeatedly making the same decision as they assume this to be rationally the best choice option e.g. travellers always choosing the same route for daily commuting to work. Game theory describes this as a *system equilibrium.* A system equilibrium is a state which remains unaltered unless influenced by external factors. Therefore, change in traveller behaviour requires external change such as closing a road to force travellers to make alternative route choices. [41] have demonstrated the existence of the *Nash equilibrium* for non-co-operative games in which there is a finite number of players, strategies and possible outcomes. The Nash equilibirum emphasises the stability within a combination of strategies in which *no single player* has the incentive to be *the only one to deviate* from the equilibrium [59]. Conflict situations are commonly analysed by determining Nash equilibria. They can be used to mathematically predict player decisions and thus, determine the anticipated outcome of a game. Formally, the Nash equilibrium can be described as follows. Let S_i be the set of available strategies for the i-th player and S the set of possible combinations of individual strategies (strategy profile). For each $s \in S_i$ there is a payoff/reward $r_i(s)$ when evaluated in a combination of individual strategies $x \in S$. Therefore, the payoff $r_i(s)$ of the i-th player depends on the choices of all players and is denoted as $r_i(s, s_{i'})$, where $s_{i'}$ represents the strategies chosen by all other players except player i. An individual's mixed strategy s_{mixed} is denoted by a probability distribution on the set of available strategies S_i. This means that the individual strategies of the i-th player $s \in S_i$ are selected based on probability p_i. A pure strategy s_{pure} is a special form of a mixed strategy, for which probability $p_i = 1$. A Nash equilibrium $n \in S$ is a special combination of individual strategies for which $r_i(n) > r_i((s_1, ..., s_{i-1}, s_i, s_{i+1}, ..., s_n), s_{i'})$ for all i, where $s_{i-1}, s_{i+1} \in S_i$ are strategies other than s_i available to player i, and $s_{i'}$ represents the strategies chosen by all other players except player i.

The Braess paradox demonstrates *inefficiency of equilibria* [56]. The outcome of the overall system can be worsened as a result of the self-interested behaviour of players (see [60] for a discussion). An important indicator for measuring the inefficiency of equilibrium situations is the *price of anarchy* [61]. This indicator measures the inefficiency that arises when individuals pursue their self-interest without coordination. In particular, looking at the worst-case equilibrium to determine how much worse the system may perform compared to an ideal scenario in which the behaviour of individuals is aligned to achieve what is best for the common good. Formally, the price of anarchy is defined as the ratio between the optimal global outcome and the worst case of possible equilibria.

$$Price of anarchy = \frac{optimal\,global\,outcome}{worst\,case\,equilibrium} \tag{2.1}$$

In addition to this, the *price of stability* looks at the best-case equilibrium and thus how close the performance of the system can get when individuals act according to their self-interest in comparison to the ideal scenario. Hence, the *price of stability* can be formally defined as follows [60].

$$Price\,of\,stability = \frac{best\,case\,equilibrium}{optimal\,global\,outcome} \tag{2.2}$$

2.2.5 Mechanism Design

The Braess paradox demonstrates that systems can potentially end up in suboptimal equilibrium states. Mechanism design aims at changing the game theoretic structure of the system to move equilibria to more favourable states. For this purpose, it is important to look at the *rules of the game*. Rules of the game refer to the set of instructions or specifications that govern how a particular game is played. These rules define the structure of the game, the available set of action strategies, the information that players have, as well as the mapping from strategy choices to outcomes (see [45] for a formal definition). In game theory, these rules are also referred to as *mechanisms*. Hence, *mechanism design* is a specific branch of game theory that deals with changes in the regulatory framework by superordinate entities (e.g. institutions [62]) to achieve a desired global outcome (see [63]–[65]). Implementation of rules and incentives can be used to reallocate resources and thus guide self-interested player behaviour towards social benefit. In doing so, positive behaviour is promoted while negative behaviour is impeded. For example, road traffic is responsible for a significant share of greenhouse gases that have a negative impact on the environment. The environment is a shared social good while personal mobility represents strategy choices of an individual. The interactions between decisions on personal mobility and the shared environment may cause externalities that influence the payoffs and outcomes for all players involved. Individuals are self-interested and prefer transportation that is affordable, comfortable and fast. As a consequence, current traffic has reached the state of a system in overload (e.g. traffic jams or perpetual lack of parking space) in which too many individuals travel in private vehicles. [57] has demonstrated the use of tolls to resolve the Braess paradoxical equilibrium and redistribute the traffic load. Other approaches look at promoting alternative forms of mobility that avoid the emission of exhaust fumes. For example, by subsidising tickets for public transport or improving cycling routes. These measures aim at changing the perception of personal utility to induce behavioural changes in the system. However, identifying appropriate and effective interventions can be difficult. For this purpose, mechanism design theory provides conceptual tools that allow the design and evaluation of policies in a system [66]. Social Choice functions (SCF) are used to model social benefit that results from a given mechanism and thus can be used to assess the effectiveness and efficiency of a regulatory framework. [67] has discussed two important properties of SCF that mechanism design aims to achieve. The first property is referred to as *ex-post efficiency*. This property describes a mechanism in which all outcomes of the SCF are Pareto optimal (see [68]). A situation is considered Pareto optimal when it is not possible to improve the outcome of any individual without having to make it worse for someone else. This is the case when the rules of a game are designed so that maximising individual utility naturally leads to an optimal global outcome. Moreover, SCF should be *non-dictatorial*. In a dictatorial setting, the decisions of the players are irrelevant and there would always be an optimal global outcome regardless of the constellation of player choices. Research on mechanism design has looked at various scenarios to determine systemic conditions in which social benefit approximates ex-post efficiency and non-dictatorship.

2.3 Validation and Verification

When applying game theory and mechanism design concepts to model and analyse real-world scenarios e.g. in traffic, scenarios are obviously not limited to two-player games but have a significantly larger number of individuals involved in a game instead. This added complexity requires the use of computer-based simulations for the analysis and evaluation. Building simulations is a complex and error-prone process. It is therefore important that simulations are being validated and verified [36]–[38]. Theory describes different stages of validation and verification depending on model maturity

[9], [40], [69] (see Figure 2.5). While model maturity as well as the different stages of validation and verification can be considered as an iterative process [9], practical application shows that the order of validation and verification is often disregarded and may even happen concurrently. Furthermore, practical application often blends activities from different validation and verification stages, meaning that there is not always a clear separation of these stages as described in the literature. Nevertheless, theory provides a useful overview of the different activities involved in the validation and verification of simulations. In particular, literature distinguishes between the two validation steps *conceptual model validation* and *operational validation* while the term *verification* applies only to the computerised model of the simulation [69]. [69] and [70] have discussed various verification and validation techniques for application on both the overall system as well as submodels of a simulation. Due to the large number of available validation and verification techniques, it is typically not possible to apply all of them to a simulation model. Therefore, the selection of techniques may vary depending on the type of simulation as well as available resources and time constraints [71]. Furthermore, there are methods that are particularly suitable for certain stages of model development. The following sections briefly reflect on important validation and verification techniques that can be used in the relevant steps of the model development.

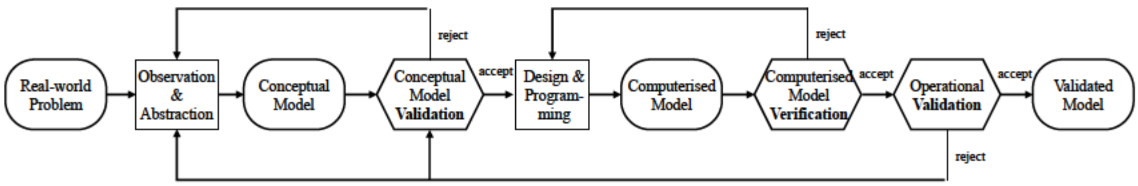

Figure 2.5: Model validation (see [9]).

2.3.1 Operational Validation

Most of the work typically associated with the validation of simulations falls within the scope of *operational validation*. This stage examines the *accuracy* of simulation output to replicate real-world data. It is important to mention that simulations can only be validated with respect to their intended purpose and that they cannot be assumed to also be valid for other purposes [72]. Furthermore, it needs to be acknowledged that simulations can only be as reliable as the data used for building them. Validation techniques used in operational validation include for example *comparison* of simulation output to either historical data or output of other simulation models using visualisations and statistical methods (e.g. [73]–[75]). The comparison of simulation output and historical data does not necessarily have to be limited to values of specific performance indicators but may also look at trends and patterns in the system. Other techniques *examine the output behaviour* by performing extreme conditions tests, or changing input variables and looking at effects on output data for already known and plausible cause-effect relations (*sensitivity analysis*). Rigorous operational validation requires both *simulation output* as well as *output behaviour* to be tested over multiple runs under varying experimental conditions. A major difficulty in operational validation lies in obtaining the necessary validation data. Apart from availability, this data also has to ensure comparability as well as sufficient representativeness. Comparability refers to whether the situational conditions in the simulation match the circumstances in which the real data was collected. Representativeness can be understood as a sufficiently broad range of data that allows conclusions to be drawn about the system as a whole from the data sample. Given that operational validation takes place at the very end of the model development process, it needs to be assumed that inconsistencies may be caused by any of the preceding activities involved in building the simulation model. In particular, errors may occur during the creation of the conceptual model or during the implementation of the computerised model.

2.3.2 Conceptual Model Validation

This stage refers to activities that ensure that the conceptual model appropriately reflects the processes and characteristics of the real world. Creating a conceptual model is a cognitive process and therefore the conceptual model only exists intrinsically within the mind of the modeller [8]. Validating the conceptual model requires some type of explicit representation (e.g. flow charts, cause-effect graphs, list of assumptions), and is based on two actions: First, theories and assumptions used for creating the conceptual model need to be thoroughly examined for completeness, consistency and plausibility. This can be achieved by analysing different real-world situations of a problem entity and crosschecking model assumptions against observational data. Second, it is important to ensure that the mathematics and logic used to formalise the conceptual model appropriately reflect the previously defined theories and assumptions. Validation techniques used for this purpose include *face validation* and *traces* [9]. During face validation, domain experts are involved to discuss and evaluate the conceptual model. Traces describe the process of following the behaviour of specific entities throughout the model and submodels in order to confirm the correctness and accuracy of the applied logic and mathematics [69].

2.3.3 Computerised Model Verification

The computerised model is a specific explicit representation of the conceptual model that has been implemented as an executable piece of code. The use of programming languages and simulation packages may lead to deviations in the implementation of the conceptual model. Ensuring that the implementation appropriately reflects and executes theories and assumptions from the conceptual model is referred to as computerised model verification. Verification techniques include *static* and *dynamic testing* [69]. Static testing examines the system design (e.g. generated UML diagrams) and correctness of algorithms (e.g. unit tests). In contrast, dynamic testing checks the computerised model from a *behavioural perspective* for example by executing simulations under different conditions. Note that in this step, the output of the computerised model is not tested for whether it matches real-world data as that is part of operational validation. Instead, the output of the computerised model needs to align with the theories and assumptions of the conceptual model. Other techniques used for dynamic testing include animation (visually animating simulated activities), traces or face validation.

2.4 Traffic Simulation

Computer-based simulations are an established means to research systemic patterns in traffic. They have been used to estimate the effects of potential changes to the traffic system. This is particularly relevant when testing in vivo is at least expensive or even infeasible. For example, long-term changes such as the extension of infrastructure have long implementation times and require a substantial deployment of resources. It is therefore important for potential interventions to be carefully scrutinised in advance. According to [28], the evolution of traffic simulations parallels the technological advancements of digital computers: Starting in the 1950s, digital computers were still in an early stage and have been primarily developed for the military. As a result, access to these computers as well as the available computing capacity was highly limited. Most of the work during this time was spread across different disciplines such as *mathematics, economics, aerospace* and *computing*, and primarily focused on theory. This produced important groundwork and created the new field of *transportation engineering*. With the 1960s, computer technology improved and became more affordable which made it easier for researchers to get access to computers. During this time, early simulation languages (e.g. SIMCSCRIPT) have been developed as well as the first implementations of applied traffic simulation. In the 1970s, computers featured more computing capacity which allowed simulations to cover larger road networks and to simulate traffic in more detail. 1980 marks the spread of personal computers which led to even more researchers being able to work on traffic simulations. As a result, the first simulations for rural areas appeared during this time. Since then, traffic simulation has had a wide spectrum of applications that each deal with different aspects of the transport system. Over the years, research has continued to find new methods to improve the simulation for various problem scenarios and thus incrementally produced more elaborate simulation models. This section gives a brief overview of the different types of simulation models and reflects on the use of multi-agent techniques in computer-based traffic simulation.

2.4.1 Types of Simulation Models

Traffic simulations can focus on different aspects of the transportation system depending on the research objective. Therefore, the level of detail considered in those simulations may differ significantly. In the literature, simulation models are often divided into four categories [76], [77]:

1. *Macroscopic simulations* focus on traffic flow modelling based on high-level mathematical models. This type of simulation can be used for the analysis of wide-area systems in which no detailed modelling is required, e.g. the simulation of motorway traffic. Given the low level of detail, macroscopic simulations are relatively fast and require less computing power.

2. *Microscopic simulations* focus on modelling individual entities based on a high level of detail. Possible entities include travellers, vehicles, traffic lights, etc. This type of simulation is often used for the analysis of urban traffic. It is possible to analyse both macroscopic and microscopic aspects (e.g. traffic lights algorithm, multimodal traffic) of the system. Consequently, microscopic simulations may result in longer computing times.

3. *Mesoscopic simulations* are a mixture of macroscopic and microscopic simulation models. Traffic entities are modelled at a higher level of detail than macroscopic approaches, however, the interaction and behaviour of the individuals are less detailed.

4. *Nanoscopic simulations* are even more detailed than microscopic approaches. This type of simulation is applied in the field of autonomous driving, in which internal functions of the vehicles such as gear shifting or vehicle vision have to be examined.

The level of detail determines which aspects of the transport system are covered. Such differences are also reflected in the data required for modelling. The use of real-world data should increase the realism and accuracy of simulations. However, researchers need to be aware of the purpose of their simulation and choose a simulation model that supports the required level of detail for dealing with their research objectives (see [78] for a discussion). Going into more detail than necessary can make a simulation model complex and also require more input data. For example, macroscopic, mesoscopic, microscopic and nanoscopic simulations require at least two types of input data which may vary in the considered level of detail:

- **Travel Demand**: This type of input data defines the requirement for travel and thus the resulting traffic volume between locations. This can be modelled using either activity- or trip-based approaches. Depending on the selected modelling approach different input data are required. For example, activity-based approaches use information from census and behaviour surveys to generate daily activity schedules of individuals and thus create the need to travel. In contrast, trip-based approaches make use of origin–destination (OD) matrices which require no information on the daily schedules of individuals and thus allow for a more abstract representation of traffic. However, trip-based approaches can also consider different levels of detail. At the macroscopic level, this may be modelled through distributions of vehicles moving between larger areas, e.g. the number of vehicles per hour moving between a group of towns. This information may come from traffic surveys or census data (e.g. giving the number of daily commuters between two towns). However, for microscopic simulation it becomes necessary to differentiate between individual vehicles. Rather than moving between two towns, demand may be modelled in the form of specific journeys from one address to another address for a specific reason (e.g. commuting or shopping). Within mesoscopic simulation, journeys are typically simulated from a general location to a specific address, for instance commuter journeys that begin from a town but travel to a specific employer's address. In order to simulate at the microscopic levels, high-level OD matrices need to be modelled in more detail, with entries for specific addresses. For specifying demand as specific journeys, analysing travel diaries and census information gives insight into the travel habits of individuals. Nanoscopic simulations often focus on a smaller geographical area in which demand may be represented by those journeys that are completely within the simulation as well as those that either pass through the simulation or only start/end within the area. Demand is likely to be specified as individual journeys, once again best specified using census or travel diary data.

- **Infrastructure**: This type of input data comprises information on the road network. At a fundamental level, the road network is a graph of nodes and edges that represent junctions and roads respectively. The amount of detail required at the macroscopic level is minimal, possibly denoting that a route between two towns exists and its capacity/travelling time and only taking into account trunk routes. When using simulations for which greater levels of detail are required (e.g. microscopic and nanoscopic) it becomes necessary to include lower-capacity roads and intermediate junctions in the road graph. At the microscopic level, the graph will need to contain information such as lane capacities, and junction types. At this level, the difference made by features such as traffic signals, turn restrictions or lane closures may radically affect the outcome of the simulation. OpenStreetMap (OSM) [79] can provide a detailed source of road network data that can be applied at most levels of simulation.

2.4.2 Multi-Agent Modelling

State-of-the-art research on traffic simulation has shown a growing interest in examining traffic as an emergent phenomenon, rather than a problem that can be modelled from a global perspective. Emergent traffic models assume that global system behaviour results from the interactions between the personal behaviours and preferences of a large set of individuals [24]. Therefore, the application of multi-agent models can be particularly suitable for the simulation of traffic. They can be positioned as microscopic simulations that can also be used for more coarse-grain research purposes (mesoscopic and macroscopic). [80] provide a description of common structures found in agent platforms that are designed for the simulation of traffic.

Historically, research on multi-agent systems has focused on the collaborative aspect of distributed systems. Agents can be both hardware or software systems that perform tasks in complex, dynamically changing environments [81]. In the 1990s there has been an increasing interest in the concept of *intelligent* software agents. As a consequence, aspects such as decision-making, interaction and autonomous behaviour became relevant. With this, the term agent has evolved into what it is presently known for (see [82]–[84]). One of the most referenced definitions was given by Wooldridge. Wooldridge describes software agents to be *closed computer systems that are situated in some environment, and that are capable of autonomous action in this environment in order to meet their designed objectives* [83]. Based on this and the aspects described above, the following characteristics are associated with the concept of intelligent software agents.

- **Autonomy.** Intelligent software agents are able to act autonomously. This describes the ability to independently make and execute decisions without being controlled by an external entity. The decisions are purely based on internal properties of the agent and agents do not have to be permanently controlled.
- **Deliberation.** Intelligent software agents are modelled to have goals that they proactively try to achieve through the execution of actions.
- **Reactivity.** Intelligent software agents always exist within a defined environment. They are able to perceive their surrounding and to react to situational changes.
- **Interaction.** They are able to interact and communicate with other agents as well as other objects in the surrounding environment.

Multi-agent systems are used in various fields of application (see [85]). They are an established means for the *construction of synthetic worlds* [86]. The creation of these artificial universes can be used to analyse interactions. Based on this, agent-based systems have been applied to the simulation of complex systems [86] such as formalised games [87]. Synthetic worlds based on agent models can help overcome the limitations of classic game theory approaches. For example, agents that perceive information about their environment through sensors allow the implementation of decentralised knowledge which makes decision-making more realistic. This as well as the autonomous and goal-driven behaviour of intelligent software agents make agent models particularly suitable for the representation of individuals in road traffic. For example, travellers can be modelled as agents that interact and perceive information about their environment through sensors, allowing for the implementation of decentralised knowledge and thus autonomous behaviour based on situational conditions. This approach to modelling individuals is the key distinction of multi-agent approaches from other types of simulation models e.g. cellular automata [88].

There are different approaches to the implementation of agent models in practical applications. [89] provide a discussion of various implementation approaches. Agent models differ primarily in how they structure agent knowledge and in their use of this knowledge for modelling decision-making behaviour. Literature distinguishes between *quantitative* and *qualitative* decision-making [51], [90]. Quantitative approaches match concepts from game theory in which mathematically modelled utility functions are used to determine the optimal course of action. In particular, *Bayesian networks* can be used to model one-shot decision problems (see [91]) while *Markov decision processes* can be used for more complex problems that require a series of decisions to be taken subsequently (see [92]). The representation of decisions and their reasons becomes significantly complex in quantitative models when decisions require a broad knowledge of the world [46].

In comparison to this, qualitative approaches soften the assumptions of traditional quantitative decision-making and a more abstract non-numerical representation of decision preferences is chosen [93]. Decision-making or the transformation of non-numerical preferences into specific actions requires alternative reasoning mechanisms [94], [95]. For example, non-numerical preferences can be represented using *relations*. In this context, *dominant relationships* are an important type of relations. [96] demonstrated formalisation of qualitative dominance based on the *Surething*-principle formulated by [52].

Bringing together the benefits of both quantitative and qualitative modelling may lead to rich agent models with a broad knowledge of the world. In this context, *mental-level models* from qualitative decision theory should be noted [97], [98]. Mental-level models describe machines and software programs to possess an abstract level for representing mental attributes such as *beliefs, knowledge,* and *preferences.* Brafman and Tennenholtz were one of the first that formalised a method for representing the state of an agent using mental attributes [99]. Knowledge abstraction into a separate level comes with advantages. Mental-level models provide a uniform basis for the comparison of agent behaviour which helps theoretical analysis [99]. Reference is given to the *computers as believers* paradigm which allows for abstract analysis of knowledge representation schemes [100]. Furthermore, mental-level models enable a more intuitive modelling which is an important aspect in design validation.

Belief-Desire-Intention (BDI) architectures and their derivatives are implementations of mental-level models [101], [102]. The BDI model enables software agents to perform action decisions (intentions) on the basis of defined goals (desires) and their modelled knowledge of their external world (beliefs) [101]. More specifically, *beliefs* represent all information about the current state of the world in which the agent currently resides. This information is stored in a knowledge base that contains data about the current environment, the internal state of the agent as well as background knowledge required for the decision-making process. Beliefs are constantly updated by the perception of the agent. Apart from this, desires describe the main objectives of the agent which fundamentally affect agent behaviour. Desires are a crucial part of intentional behaviour, as agents are not prompted to take any further actions without targets. A modelling approach using target-oriented behaviour enables the modelling of continued pursuit of the selected target by alternative actions in case of a failed action. The set of alternative actions or plans is portrayed as *intentions.* Each agent possesses a variety of plans from which the agent can select an action to achieve a set target. The BDI model separates the selection process of a plan from its execution process. Consequently, the BDI model is well suited to model actors in traffic scenarios: not only destinations of journeys but also optimisation goals e.g. minimal travel time and minimal emissions can be formulated as desires. Travel preferences and other parameters are potential beliefs that can be used to determine e.g. modal choices.

2.5 Summary and Conclusion of Chapter 2

In this chapter, an introduction was given to the basic notions and theoretical concepts related to this thesis. Section 2.1 covers terminology on systems, models and simulation and thus establishes important groundwork. The discussion of the modelling process in section 2.2 illustrates the different phases of maturity when building simulations, from the description of a real-world problem to the final implementation of a simulation model. In this context, this section specifically looked at game theory as it contains important contributions to the formalisation and analysis of real-world situations, and is particularly relevant for understanding complex scenarios in road traffic. For example, the effects of self-interested individuals and their decisions on social benefit as demonstrated by the Braess Paradox. Furthermore, improving traffic systems with their current constraints to achieve a more favourable state requires careful planning of measures that can bring about the necessary change. For this purpose, mechanism design covers important theory that discusses the effects of changes to the structure of a system which is particularly relevant for policy-making.

Following this, section 2.3 discusses the different aims and approaches to validation and verification of simulations, underlining the importance of ensuring the reliability of simulation models for practical applications from different perspectives. In particular, conceptual model validation looks at validating the theoretical ideas of the simulation model while computerised verification focuses on whether the theoretical model has been implemented correctly. Operational validation examines whether the results of the simulation match with observations of the real world. This is the basis for addressing the final research question of this thesis that deals with improving the ability to understand the internal mechanisms of agent-based traffic simulations when the complexity of simulation models increases.

Finally, section 2.4 provides a comprehensive overview of traffic simulations, reflecting on their historical context as well as illustrating the different types of simulation models i.e. levels of detail and the application of multi-agent models. Hence, this section provides the general context for further examining individuals and their behaviour in such simulations.

Chapter 3

A Systematic Survey on Modelling Individuals Using Agent-based Traffic Simulators

This chapter provides a systematic survey of current approaches to modelling individuals in agent-based traffic simulations. In particular, the focus is on investigating the current state of implementation of existing traffic simulators to model aspects of individuals and their behaviour. For this purpose, the next sections describe the scope and method used in this survey. Furthermore, perspectives of comparison as well as a detailed review of the simulators are given. This chapter concludes with a discussion of the challenges and limitations of simulators. An earlier version of parts of this chapter has been published in [4].

3.1 Method

Based on methods found in similar surveys (e.g. [103], [104]) a keyword search has been performed for peer-reviewed papers across the three common publication databases: Google Scholar, ACM Digital Library, and IEEE Xplore; on the title, abstract, and the main body of the papers. The selection of papers only includes publications written in English. Duplicates and irrelevant papers have been excluded from the result set. Search results are based on the three keywords: 1. Traffic Simulation, 2. Agent-based Traffic Simulation and 3. Multi-agent Traffic Simulation. The first 30 research papers from each database and each keyword were included in a backward search to identify simulators that are considered related work by the authors of the publications. Furthermore, the search was completed with findings from forum discussions in the research community.[1] It should be noted that only simulators have been considered that primarily focus on road traffic. Publications on maritime or air traffic have been excluded. In the case that multiple papers relate to the same simulator reference has been given to the paper cited the most. Based on this, the selected simulators have been analysed to obtain an overview of implemented features relevant to the modelling of individuals and their behaviour.

3.2 Perspectives of Comparison

As there are a lot of research activities that have focused on different aspects of the transportation system and vary in the considered research objective, literature research has produced a diverse set of simulators that differ in both their underlying model and scope of application. In particular, the models implemented within these simulators vary in the extent to which they employ agent technology. In this research, a distinction is made between simulators that are *non-agent-based, fully agent-based* and *hybrid simulators*. Hybrid simulators are extensions of non-agent-based simulators that have added agent capabilities for specific aspects of the simulation software. Simulators that use software agents to fully implement decision-making concepts in traffic (e.g. travellers or vehicles) are referred to as *fully agent-based.* In addition to this, there are also general-purpose agent platforms such as NetLogo [105] that can be useful for implementing lightweight experiments. This research focuses on simulators that are designed for the simulation of large-scale scenarios (e.g. simulations with a population size greater than 100.000) as these are most relevant for conducting real-world case studies. Note that general purpose platforms might not be suited by default for the

[1] see `https://www.researchgate.net/post/What-is-the-best-agent-based-traffic-simulation-tool` - (access on 13/11/2023)

simulation of large-scale scenarios in high performance computing environments but can be adjusted by modifying their configuration. The following list of simulators has been found in the scope of this survey:

- *Fully Agent-Based*: MATSim [106], ITSUMO [107], MovSim [108], MASCAT [109], MATISSE [110], POLARIS [111], AgentPolis [112], OPUS [113], MOSAIIC [114], MARS [115], SimMobility [116], SITRAS [117], ArchiSim [118], SEMSim (CityMOS) [119], JTSS [120], Megaffic + XAXIS [121], SD-Sim [122], SM4T [123], VCTS [124], SIMTUR [125], MUST [126], CAMiCS [127], OpEMCSS [128], DEFACTO [129], MAGE [130], CityScape [131], BAE Systems [132], AITSPS [133], SeSAm [134], IMAGES [135], Mobiliti [136], CUPSS [137], KLMTS1.0 [138], CARLA [139], AgentStudio [140], ILUTE [141], SIMULACRA [142], TransWorld [143], SUMMIT [144]

- *Hybrid*: ATSim [145], FastTrans [146], SUMO+JADE [147]

- *Non-Agent-Based*: TRANSIMS [148], SUMO [149], OpenTraffic [150], [151], PACSIM [152], PTV VISSIM/VISUM [153], GETRAM/AIMSUN [154], PARAMICS [155], MITSIM [156], FreeSim [157], TSIS/CORSIM [158], VATSIM [159], DRACULA [160], RENAISSANCE [161], SimTraffic [162], DynaMIT [163], DYNASMART [164], MITSIMLab [165], CUBE Voyager [166], PELOPS [167], TransModeler [168], Dynameq [169], CORFLO [170], SIMSCRIPT II.5. [171], CTSP [172], CityMob [173], VanetMobiSim [174], FIVIS [175], THOREAU [176], GENIVI [177], SLX [178], SALT [179], SIM-ENG [180], KAIST [181], UMTSM [182], SES/MB [183], SISTM [184], INTERGRATION [185], MATDYMO [186], TRANSYT [187], CONTRAM [188]

This survey focuses on *fully agent-based* and *hybrid simulators* as the objective of this research is to address the issue of modelling individuals. Modelling individual behaviour deals with different modelling aspects depending on the simulated level of detail (see Chapter 2.4.1) that is relevant for the scope of application. In particular, the time perspective considered by a simulated problem scenario determines the relevance of aspects to be included in the modelling. For example, when modelling individual behaviour in the context of mobility, decisions about workplace and residency are more relevant when research questions address *long-term* matters such as land use. In contrast, *short-term* behaviour typically involves modelling individuals on a micro-scale, thus dealing with spontaneous interactions, lane changing, or acceleration and braking. Aspects on *mid-term* behaviour are in between and include for example decisions on prejourney planning such as route choice or selection of travel modes. Another important aspect related to mid-term behaviour is *demand modelling*. Traffic simulations have modelled travel demand based on either trip- or activity-based methods. Trip-based demand modelling typically uses *origin-destination (OD)* matrices to specify the amount of traffic between two locations. Input to OD matrices may include static values or probability distributions. An alternative approach to OD matrices is *location-specific probabilities (LSP)*. Locations are assigned a pair of probabilities for the amount of travellers starting and stopping at this location. In this case, it is assumed that travellers are moving in space with no route specification. In comparison to this, activity-based approaches model demand by generating individual sets of activities that reflect the time schedule of a simulated traveller individual (e.g. bringing children to school, going to work in the office, picking up the children from school and going grocery shopping before returning home). In this case, OD matrices are a consequence of generated activity schedules.

Considering the diversity of aspects in modelling individual behaviour, it is important to review the simulators in the context of their application area. [189] have discussed relevant areas of interest in mobility to which a significant amount of research and funding is currently devoted. Based on this, the three areas of application: 1. *Social Dilemmas in Resource Utilisation*, 2. *Digital Connectivity* and 3. *New Forms of Mobility*; will be considered in this review. Given the large number of available simulators, it is not possible to review all of them in this research. Thus, for each application, three examples of simulators are given that have been used to model issues related to this application domain. It should be noted that areas of applications are closely connected and therefore may be overlapping. Hence, simulators mentioned as an example do not have to be used exclusively for the mentioned area of application but can be particularly suitable. In the scope of this survey, a brief overview is given of the background and technical setting of the simulators. Simulators are then examined on their ability to model aspects of individual behaviour by discussing implemented features.

3.3 How Traffic Simulators Model Individual Behaviour for Current Areas of Application

3.3.1 Social Dilemmas in Resource Utilisation

This application domain considers the issues arising from transport infrastructure being a shared resource used by many individuals, but not owned by any one of them. This means that the use of transport infrastructure by one individual potentially creates externalities that affect other individuals, e.g. congestion and pollution [57]. Better public transport and shared mobility services are intended to relieve the traffic load on roads, while electrification of vehicles is seen as a means to reduce exhaust fumes and environmental pollution. This creates new questions as to what effects will be achieved in the short term as well as in the long term. For example, e-mobility inevitably leads to a change in energy consumption that requires efficient planning of available resources. This section specifically looks at the three traffic simulators MATSim, POLARIS and SimMobility as examples of agent-based simulators that have already been used to simulate issues in this context. Other simulators that also fall into this category include: SEMSim (CityMOS), Megaffic + XAXIS, MUST, CAMiCS, DEFACTO, CityScape, BAE Systems, SeSAm, Mobiliti, MARS, MOSAIIC, OPUS, ILUTE, SIMULACRA.

MATSim

MATSim is an agent-based software framework implemented in Java and licensed under GPLv2 or later [190]. The project started in 2004 at ETH Zurich and is currently being developed in collaboration with TU Berlin and CNRS Lyon. The framework has a general focus and is designed for the simulation of large-scale transportation scenarios. Hence, a particular effort was made for efficient computational processing and parallelisation [191], [192]. MATSim has been used in particular to simulate energy demand planning in transportation [193], [194]. The framework consists of five components for *Initial Demand, Execution, Scoring, Replanning* and *Analysis* (see Figure 3.1) [106]. Based on the modular approach, custom components can be implemented and integrated into MATsim in order to replace or upgrade provided default operations. The first component *initial demand* deals with the modelling and generation of an initial agent population. Agents select and execute plans in the *execution component*. The *scoring component* calculates a score for every plan based on a given utility function. This score is an indicator for accomplished agent utility. The *replanning component* uses a *co-evolutionary algorithm* for optimising this utility. In contrast to an ordinary evolutionary algorithm that searches for a global optimum, the co-evolutionary algorithm is applied to evolve the set of agent plans of the travellers. The simulation cycle (execution - scoring - replanning) repeats until MATSim reaches an equilibrium and agent scores stabilise. Finally, the output data of the simulation is aggregated in the *analysis component*.

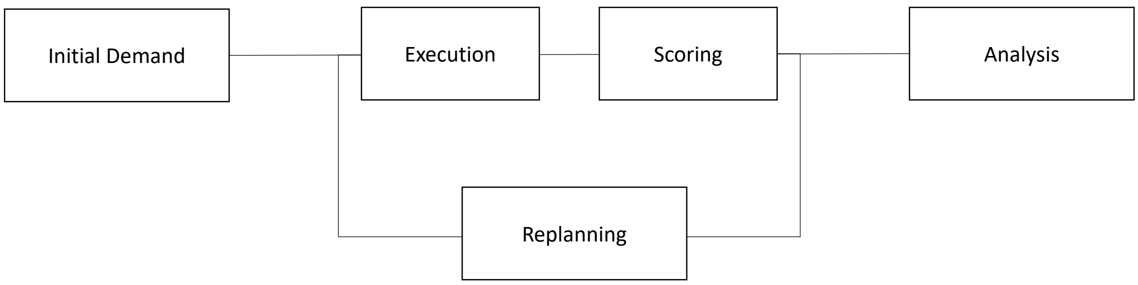

Figure 3.1: MATSim - architecture.

With regard to the modelling capabilities of the application, MATSim can be considered a mid-term simulator as scenarios are commonly modelled for single days [106]. However, there are some experiments that have demonstrated the simulation of multi-day scenarios [195]. MATSim provides two options for generating an initial population of agents which can be random or based on user input. Census information is used in order to model every traveller explicitly. The application provides a number of predefined parameters that can be configured. MATSim follows an activity-based approach for modelling travel demand. Survey data is used to generate various lists with activities that are assigned to the agents. It should be noted that travel demand changes with every iteration of the simulation as the simulation includes a replanning mechanism for rescheduling activities. Furthermore, agents possess a list of plans that contain different combinations of actions and choices. This includes choices not only about classical traffic properties such as routes and travel mode but also time scheduling. MATSim uses a discrete-choice model for implementing agent decisions [106]. Quantitative methods are used to determine probabilistic distributions for alternative actions. Agents select plans based on calculated scores from the scoring component. A higher score increases the probability of a plan being chosen (see [196]). Given the level of detail considered in the modelling of individuals, MATSim is suitable for simulating scenarios that analyse social dilemmas in resource utilisation based on the amount and types of traffic (activities and modal choices) that emerge in the system.

POLARIS

POLARIS is an open-source agent-based software framework written in C++. The project was first published in 2013 (see [197]) and is currently maintained at Argonne National Laboratory. The motivation behind POLARIS was to combine different traffic-related modelling aspects into a single framework that otherwise requires a number of separate standalone software applications. In [111], the authors of POLARIS argue that transportation research has focused on these aspects only in an isolated manner. However, the simulation of complex systems requires a combined method. Early attempts to integrate the isolated models into a unified system have shown that resulting solutions are either inflexible, non-modular or inefficient. Based on this, the authors describe a need for a unified solution that enables interoperability between the isolated models. The POLARIS framework has been proposed to address this issue [111]. POLARIS focuses on large-scale transportation scenarios and has been used to analyse the energy consumption of current and future vehicle technologies [198], [199]. The framework provides a set of tools that can be used for the development, execution and review of the simulation. The system architecture is structured using a layered approach (see

Figure 3.2). Aspect-specific sub-components are assigned to a layer depending on the level of modelling detail. This ensures abstract concepts which are commonly used across different variations of traffic simulation models to be less likely to change. Instead, users are supposed to make research-specific customisations on a more detailed level. This creates reusability of frequently used modelling aspects. Based on this, layer 0 is the most abstract layer of the POLARIS framework. Layer 0 contains a set of core libraries such as the discrete event engine which is responsible for handling agents. Simulations are performed by executing a list of events. In layer 1, POLARIS contains a set of fundamental extensions. This includes components for 2D/3D visualisation (Antares) or data import/export services. Layer 2 is described as an *open-source versioned repository*. In this repository, there is a set of model fragments that can be used for the implementation of custom simulation models. The provided model fragments are tested and chosen by universal applicability. Typical model fragments for example are reference implementations of well-established routing algorithms. Finally, layer 3 is described as the *user playground*. In this layer, custom components can be included in order to extend the POLARIS framework with research-specific modelling aspects. Based on the provided elements from all layers, the user can build a custom application for agent-based traffic simulation.

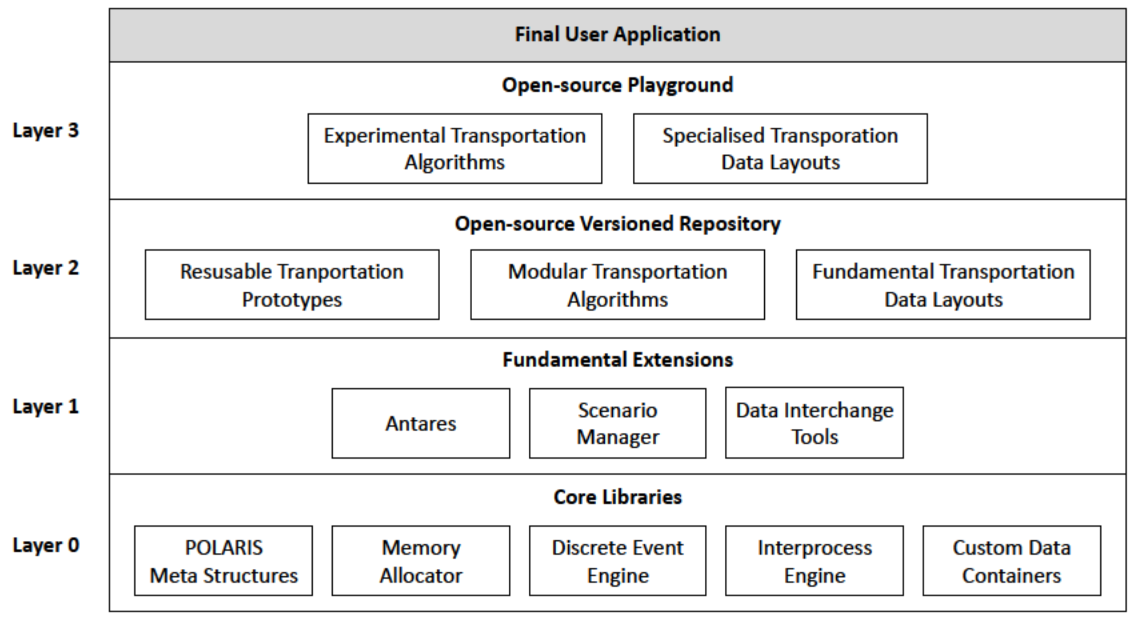

Figure 3.2: POLARIS - architecture.

With regard to modelling capabilities, POLARIS can be considered a mid-term simulator as travel decisions focus on mid-term aspects such as departure time, destination choice, route choices as well as planning and rescheduling of activities. Furthermore, POLARIS uses an activity-based approach for modelling travel demand. This approach is based on an adjusted version of the ADAPTS *(Agent-based Dynamic Activity Planning and Travel Scheduling)* model [200]. Originally, the ADAPTS model was designed as a standalone application for simulating the occurrence of travel demand patterns that result from travel planning and scheduling processes. For integration into the POLARIS framework, the ADAPTS model has been reorganised in order to match the agent paradigm. This resulted in a separate *activity planning* agent which as an extension to the traveller agent models the traveller's cognition of the activity planning process. This illustrates the applied structure for modelling other types of behaviour in POLARIS as a central traveller agent is composed of a set of subagents which each extend the traveller agent with cognitive capabilities for specific behavioural aspects. For example, these include agents for perception, movement coordination or routing. In comparison to MATsim, this approach considers a more detailed modelling of individuals allowing for easier extension of short-term behaviour. This can be useful when energy consumption needs to be determined more precisely e.g. when simulating the energy impact of acceleration and braking of autonomous vehicles to identify frequent nodes for charging stations.

SimMobility

SimMobility is a simulation platform written in C++ and published under an own open-source license. The project has related publications since 2015 and is currently developed at SMART (Singapore-MIT Alliance for Research and Technology) [201]. The simulator integrates a set of aspect-specific models relevant to the transportation domain that allows simulation on different time scales (short-, mid- and long-term) [116]. For example, aspect-specific models include land use, demographic movement or interactions related to transportation and communication. The platform focuses on modelling effects on traffic infrastructure, transportation services and the environment. This allows for the simulation of alternative planning options specifically with regard to technology, policies and investment. SimMobility has been used to simulate the effects of commercial vehicles and mobility services on the use of infrastructure [202], [203]. The system architecture of SimMobility is structured in three components and follows a multi-level approach based on the time aspect. Each component simulates a different perspective (see Figure 3.3). The first component is the *Long-term (LT)* simulator. This component deals with generating and updating the agent population. The LT simulator particularly simulates long-term aspects such as house location and car ownership, but also other long-term effects such as changes to the environment can be simulated in this module. The second component is described as the *Mid-term (MT)* simulator [201]. This component is primarily designed for the simulation of agent behaviour in time scales of minutes and hours. This refers to *high-level* travel decisions such as route choice or modes of travel. The *Short-term (ST)* simulator is the last component in the multi-level architecture which is a microsimulator based on MITSIM that has been extended with agent capabilities. A special characteristic of this architecture is that each component can be used as a standalone application. All simulators share the same database so that simulated individuals exist across all simulation levels simultaneously.

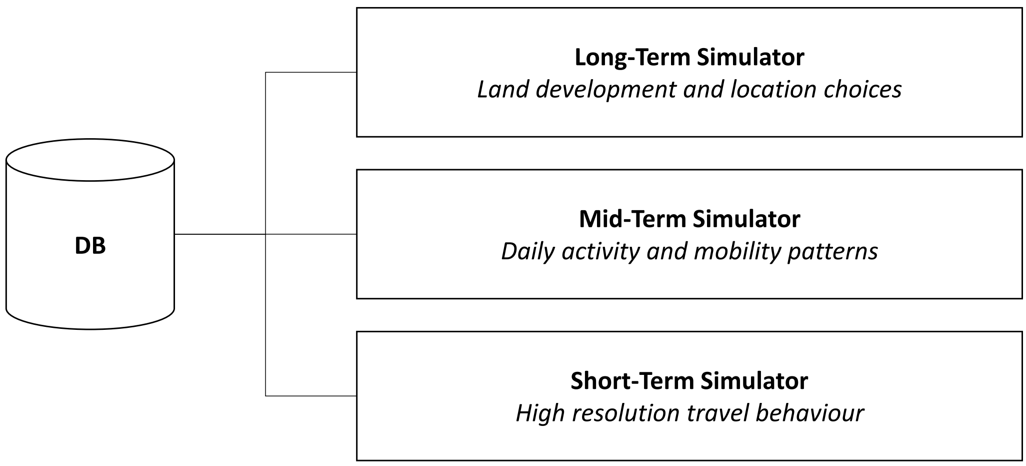

Figure 3.3: SimMobility - architecture.

With regard to the modelling capabilities, SimMobility covers all time perspectives (long-, mid- and short-term) considered in this review and therefore is particularly flexible and powerful. Modelling aspects are distributed across the three sub-components but are brought together into an individual using one database. SimMobility follows an activity-based approach for modelling travel demand [116]. For each simulated day, the MT simulator generates a list of activities that include information on destination, departure time, route and mode choice. This approach has been integrated with methods of trip-based demand modelling as generated activities are aggregated to create origin-destination matrices that can be re-calibrated. Agent decisions such as route choices are based on a probabilistic model which is similar to the MATSim approach [204]. The ST simulator also includes a mechanism that enables day-to-day agent learning to update the agent knowledge [116]. Based on these modelling capabilities, SimMobility is probably the most flexible and powerful approach in this area of application with regard to the modelling of individuals as it allows researchers that are uncertain about the required level of detail in modelling individual behaviour to easily adapt.

3.3.2 Digital Connectivity

The second area of application examines the effects of digital transformation on the mobility sector, specifically the real-time capture and analysis of traffic information. This has been used in digital traffic control systems e.g. intelligent transportation systems (ITS) to achieve higher levels of transportation safety but also to improve navigation by providing real-time information on parking and traffic jams. This section examines the integration of SUMO and JADE, ITSUMO and MATISSE as examples of agent-based simulators that have been used for research in this area of application. Other simulators related to this application include: SITRAS, ArchiSim, SM4T, SIMTUR, OpEMCSS, IMAGES, MASCAT, TransWorld.

An Integration of SUMO and JADE

SUMO (Simulation of Urban MObility) is a software framework for microscopic traffic simulation written in C++ that is licensed under EPL 2.0. A first version of the project was published in 2001 and created by the German Aerospace Center (DLR) [149]. Since then, SUMO has been accepted by a wide community. The project was motivated by the necessity for an appropriate open-source solution as other projects which are now open-source, were difficult to obtain at that time [106]. Traffic applications were mainly used as black boxes with no options to examine the underlying simulation model [149]. Thus, researchers were restricted by the given parameterisation and modelling with no option to implement custom ideas. The SUMO approach is non-agent-based but has been integrated with the *Java Agent Development Framework (JADE)* (see [147]) in order to make simulations compatible with recent agent technologies [205], [206]. JADE is an open-source software framework licensed under LGPLv2 that is used for the implementation of agent-based applications. This combination of SUMO and JADE has been used for simulating and assessing the effects of traffic control systems [206], [207]. [205] have implemented a software connector that enables communication between the two software environments. This connector is referred to as *TraSMAPI (Traffic Simulation Manager Application Programming Interface)*. From the SUMO perspective, the TraCI API is the central component for the integration of SUMO and JADE [205]. TraSMAPI communicates with the TraCI API and acts as an intermediary. Although the project focuses on the integration of SUMO and JADE, TraSMAPI is abstracted to be able to handle various simulators besides SUMO (see Figure 3.4). This makes it possible to compare the results of different simulators. The combination of SUMO, JADE and TraSMAPI can therefore be termed as an *Artificial Transportation System (ATS)* which is an extension of traditional modelling and simulation approaches with the ability to integrate different simulation models in a virtual environment [208].

With regard to the modelling capabilities, the combined approach is suitable for mid- and short-term simulations as modelling aspects include selection of travel modes but also micro-behaviour such as lane changing. JADE agents represent drivers that are linked to vehicles in SUMO. A separation of strategic and tactic-reactive agent behaviour has been implemented with two layers which is also referred to as the *delegate-agent concept* [209]. It can be understood as a separation of cognitive and reactive actions from the executing driving tasks [205]. The strategic layer deals with the collection and processing of information from the surrounding environment. Based on this informa-

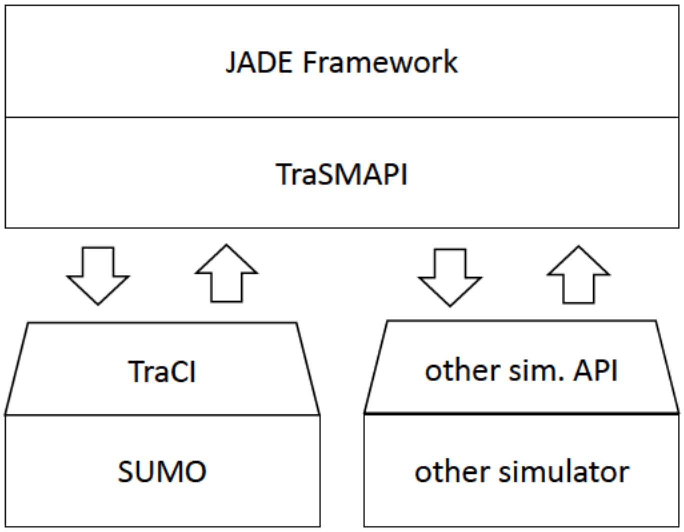

Figure 3.4: TraSMAPI - architecture.

tion the agent chooses its travel route, also in the strategic layer. In the tactic-reactive layer, driving-related behaviour such as acceleration, braking or lane changing is implemented. Based on the functional requirements of the two layers, the strategic layer was kept in JADE whereas the tactic-reactive layer was realised in SUMO (see Figure 3.5). The original SUMO package provides two options for demand modelling which can be trip-based using an origin-destination matrix [77] or using an activity-based approach. Agent decisions are based on a probabilistic model but can be extended using the TraCI API. [205] have demonstrated the application of reinforcement learning techniques to model adaptable knowledge representation. Given the microscopic level of detail in modelling individual behaviour, this application is suitable for simulating scenarios that analyse the effects of traffic control policies on the driving behaviour of individuals e.g. examining the perception of digital and analogue traffic signs.

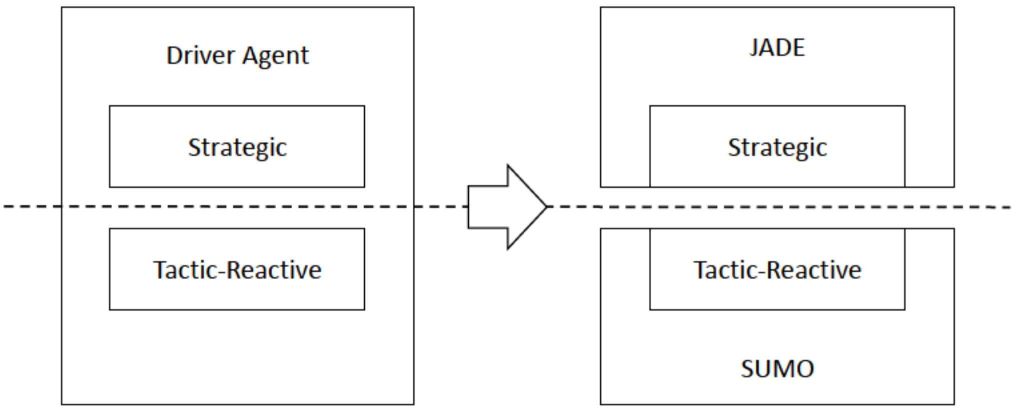

Figure 3.5: Delegate-agent concept.

ITSUMO

ITSUMO (Intelligent Transportation System for Urban Mobility) is an open-source agent-based traffic simulator written in C++ and Java. The simulator was first presented in 2006 by UFRGS (Federal University of Rio Grande do Sul) and since then has been continuously refined and advanced [107], [210]. Apart from the similarity in name, there is no direct link between ITSUMO and the previously described SUMO project. As the creators describe, ITSUMO was developed out of the lack of customising options in available simulation tools, as most of the existing solutions were developed for specific purposes. Other drawbacks described are for related simulation tools to not being fully agent-based, for them to rely on strong simplifying assumptions, or deficiencies with regards to their demand planning options [107]. Thus, the ITSUMO approach is fully agent-based and aims at addressing the deficiencies mentioned above. ITSUMO has also been applied for the simulation of route choice scenarios. However, the primary focus of the application is on traffic control. For example, ITSUMO has been used for testing traffic light algorithms [107], [211].

The system architecture is structured in five components [211], [212] (see Figure 3.6). The first component is a *database*. This database contains information about the geographic traffic network as well as other data used in the simulation (e.g. insertion rate of vehicles or origin and destination of the drivers). The second component is described as the *simulation kernel*. This component accesses data stored in the database, executes the simulation and manages agent interaction. The system architecture also includes a separate *control* component in which traffic-related control entities (e.g. traffic lights) are implemented. The control component passes information to the simulation kernel to provide instructions for simulated control entities. Finally, the results of the simulation are *output* in a separate component. For this, sensors and detectors are used during the simulation in order to collect relevant data such as travel times, average speed, etc. The output module provides two visualisation options for both, a microscopic and macroscopic view of the simulation. If the visualisation is not used, simulation data can also be output as files.

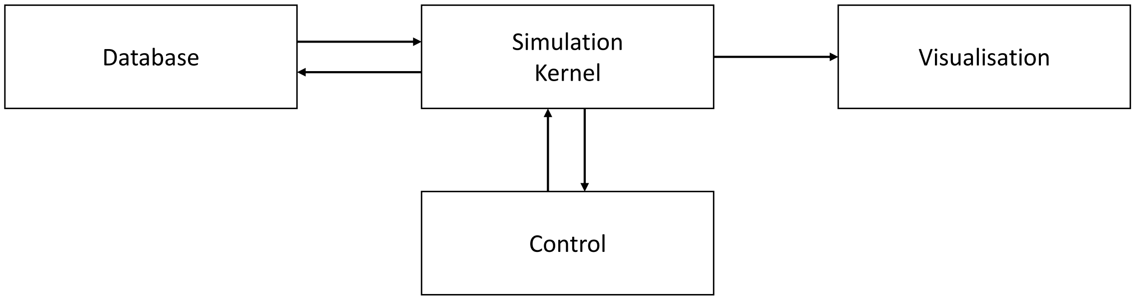

Figure 3.6: ITSUMO - architecture.

With regard to the modelling capabilities, ITSUMO can be considered a mid-term simulator that focuses on control and assignment of travel demand. Therefore, travel decisions refer to the level of route choice as well as its spontaneous replanning. Agents can either replan at every intersection or in case of a delay during the journey. ITSUMO follows a trip-based approach for modelling travel demand. Travel demand can be modelled using an origin-destination (OD) matrix or by generating a synthetic demand using uniform probabilities for a set of locations (LSP). For each combination of origin and destination, vehicles are generated and a route is determined. The application is particularly suitable for simulations that deal with ITS as it provides specific interfaces for implementing control measures and the driver reactions that are related to them.

MATISSE (DIVAs 4)

MATISSE is a large-scale agent-based simulation platform written in Java [110], [213]. The simulator has been released by UTD MAVS (University of Texas at Dallas) for non-commercial use under GPLv3 using the name *DIVAs 4*. Early work related to the project has been published since 2004 during a time when only a few fully agent-based approaches existed [214]. Within this set of fully agent-based simulation models, the creators of MATISSE criticised the lack of core agent mechanisms such as sensing, diverse communication types, etc. The project has been developed to overcome these deficiencies. MATISSE specialises in the simulation of scenarios related to traffic safety. The MATISSE architecture is structured in three layers (see Figure 3.7) [110]. The first layer is described as *MATISSE Control and Visualisation Module*. It includes a control

GUI for parameterisation and configuration of the simulation model. Furthermore, 2D/3D visualisation is implemented in this layer. Apart from this, there is a communication layer. This layer includes a *Message Transport Service* that acts as a controller in order to enable communication between the user interface and the simulation system. The third layer *MATISSE Simulation System* is the core element of the application. In this layer, calculations are performed in order to run the simulation. The layer is divided into three subsystems. The first subsystem is called *Agent System*. This subsystem is responsible for the creation and control of various agent types (vehicles, traffic lights, etc.). The *Agent-to-Agent Message Transport Service* handles agent communication during the simulation. The second subsystem is described as the *Environment System*. This subsystem creates and controls additional simulation elements related to the traffic environment. This includes elements such as the traffic network. A separate *Agent-Environment Message Transport Service* connects the environment system with the agent system. Finally, a third subsystem is the *Simulation Microkernel*. This subsystem handles all tasks related to the simulation workflow.

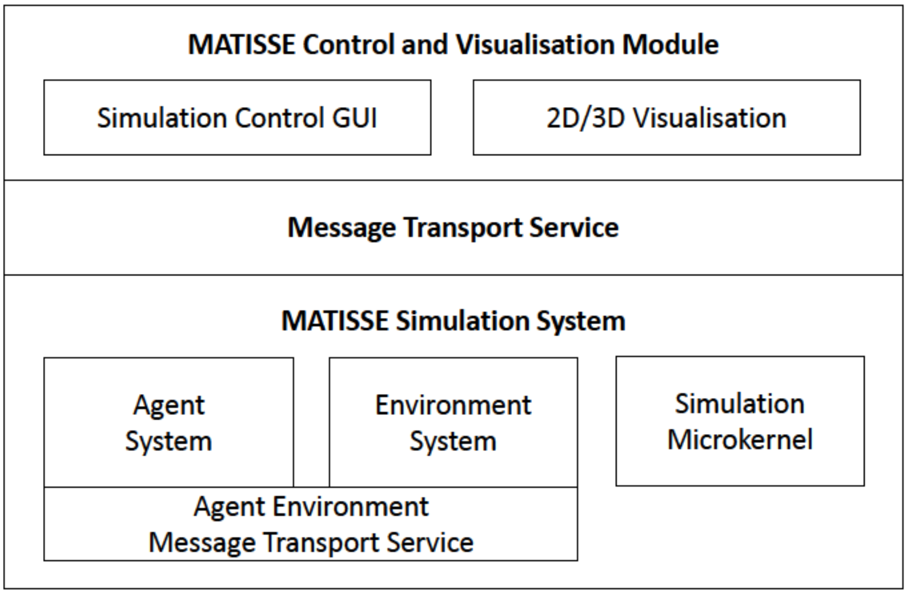

Figure 3.7: MATISSE - architecture (simplified).

With regard to the modelling capabilities, MATISSE can be considered a mid- and short-term simulator as modelling aspects focus on driver behaviour. Similar to the ITSUMO approach, MATISSE also provides an implementation for spontaneous replanning of route choices. Agent movement is based on car-following and lane-changing models, and it is even possible to model a virtual level of distraction that causes unpredicted traffic behaviour. The internal agent structure resembles a mental-level model from qualitative decision theory (see [97]) which can be useful for modelling individuals. Furthermore, mental-level models provide a uniform basis for the comparison of agent behaviour which helps theoretical analysis [99]. MATISSE follows a trip-based approach for modelling travel demand using LSP. MATISSE uses a normal distribution or a user-specified distribution in order to initialise agents for defined user entry and exit points. The application is particularly suitable for dealing with simulations on transportation safety and already provides a wide range of implementations for this area of application. The implemented mental-level structure of agents in MATISSE can be helpful for researchers that want to expand their work on modelling and analysis of individual travel behaviour.

3.3.3 New Forms of Mobility

The third area of application deals with the new forms of mobility that arise from the increasing interest in the sharing economy and the technological advances in digital connectivity. This has led to a growing portfolio of mobility services, changing the dynamics in personal mobility. For example, new mobility services such as ridesharing or -hailing as well as the achievements in the field of autonomous driving cause significant changes in the interactions between travellers and providers of mobility services, but also (autonomous) vehicles. This section specifically looks at the three simulators AgentPolis, ATSim and MovSim. They are examples of agent-based applications that focus on simulating the interaction of individuals with new mobility services or coordination dynamics of autonomous driving. Other simulators related to this area of application include: SD-Sim, VCTS, AITSPS, CARLA, SUMMIT.

AgentPolis

AgentPolis is a fully agent-based software framework written in Java and licensed under GPLv3 [112], [215]. The project was published in 2013 and created by AI center FEE CTU (Czech Technical University in Prague). The creators noted that existing simulation approaches fail to implement the ability to model ad hoc interactions among the entities of the transport system as well as the spontaneous decision behaviour that is required for this form of interaction. However, current mobility services (e.g. ridesharing) rely on frequent, ad-hoc interactions between various entities of the transport system. Hence, AgentPolis focuses particularly on the simulation of interaction-rich transport systems. For example, the simulator has been used as a testbed for benchmarking on-demand mobility services [216]. AgentPolis provides a set of abstractions, code libraries and software tools for building simulation models. The framework is structured in four main components (see Figure 3.8). The first component is described as the *modelling abstraction ontology*. The theoretical concept of this component is to separate defined modelling abstractions from implementations of specific modelling elements. It uses an ontology in order to define more general concepts of multi-agent systems that result in a tailored structure for object-oriented programming. This is particularly relevant when extending the simulation models for research-specific scenarios. Furthermore, this approach allows for the enforcement of implementations that consider the interoperability of existing and additional research-specific modelling elements in their design. The second component is a *library* of implemented modelling elements based on the given abstractions specified in the ontology. The library contains a set of modelling

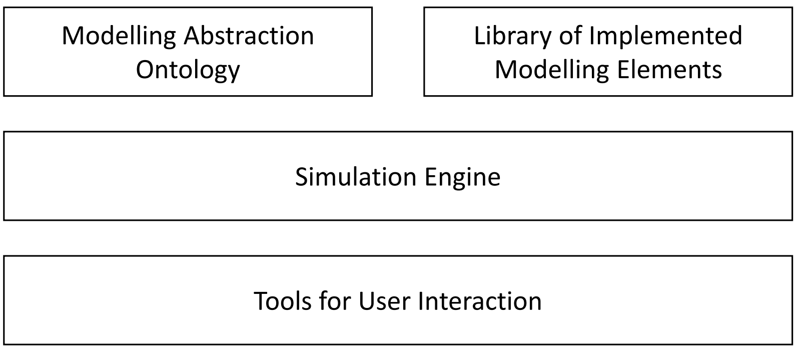

Figure 3.8: AgentPolis - architecture (simplified).

elements that represent common entities in transport systems. Apart from this, the third component can be described as the simulation engine. This component performs all calculations for running the simulation based on a discrete event model. Finally, the last component is a set of tools for user interaction, particularly for configuration and creation of the simulation model, data import, visualisation, etc.

With regard to the modelling capabilities, AgentPolis can be considered a mid-term simulator. Travel decisions refer to the level of route and modal choices. The agent structure is given by the abstraction ontology (see Figure 3.9) and defines concepts for the cognitive functions of the agent. Agents interact with objects in the environment using *sensors* and *activities*. Sensors perform queries to perceive environment objects while activities specify agent behaviour for initiating agent *actions*. Agent actions model the effects of the agent on its environment e.g. a *DriveVehicle* activity may result in a *MoveVehicle* action. [205] mention a clear separation in modelling of driver decisions and vehicle control and therefore implements decision-making of activities in a separate *reasoning module*. For this purpose, AgentPolis comes with the implementation of a multimodal *JourneyPlanner* based on a time-dependent graph [217]. [215] have extended AgentPolis with custom reasoning modules implementing different routing algorithms that were relevant to their experiments. AgentPolis follows

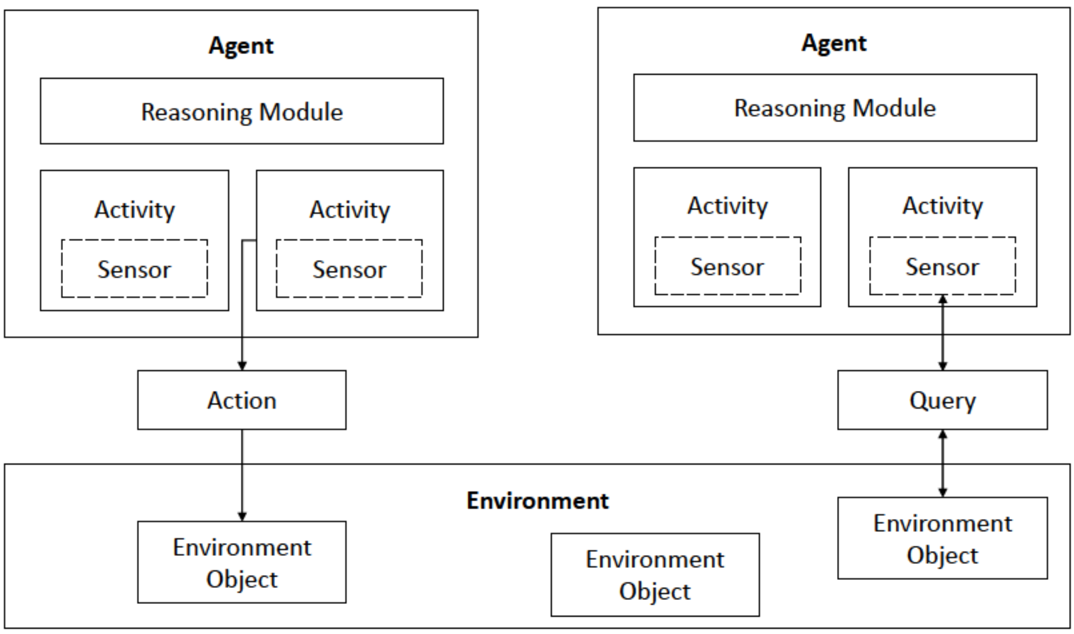

Figure 3.9: AgentPolis - agent structure.

an activity-based approach for modelling travel demand. The simulator includes a tool that generates an initial population of agents based on census data [215]. Based on the level of decision-making and implemented features, AgentPolis has been used and is suitable for simulating demand and decisions on the adoption of new mobility services [218].

MovSim

MovSim *(Multi-model open-source vehicular-traffic Simulator)* is an agent-based traffic simulator written in Java and licensed under GPLv3. The project started in the late 1990s at TU Dresden and was designed for educational purposes [108]. In contrast to most available traffic simulation tools that model specific road networks (e.g. cities), MovSim focuses on the simulation of fundamental flow dynamics. For example, MovSim has been used to simulate the effects of driver movements on traffic jams, studying the appearance of *stop-and-go waves* [219]. Because of this particular focus on flow dynamics, Movsim has also been applied for the simulation of rather unconventional scenarios such as ski marathons [220]. The simulator includes a number of reference implementations for established mathematical car-following models as described in [221]. This can be relevant to simulate lane-changing and flow dynamics related to autonomous driving. The MovSim architecture is structured in three layers (see Figure 3.10) [219]. In the *input layer*, simulation settings and parameters are defined. The user can input information either using a graphical user interface (GUI), command line or XML files. This information is forwarded to the *main loop layer*. In this layer, agent control and movement are implemented. The simulation controller continuously calculates the simulation in a loop as MovSim is based on a time-continuous model. The simulation controller primarily focuses on quantitative models. Different submodules implement the logic for aspect-specific agent behaviour such as acceleration, braking, lane-changing, etc. Two additional modules act as observers to the simulation loop in order to extract information for the *output layer*. The SimViewer module deals with information relevant to the visualisation of the simulated scenarios. MovSim includes implementation for both, 2D and 3D visualisation. Users can choose between a microscopic (cockpit perspective) or macroscopic (bird's eye) view of the simulation. If the visualisation is not used, simulated data can also be output as files.

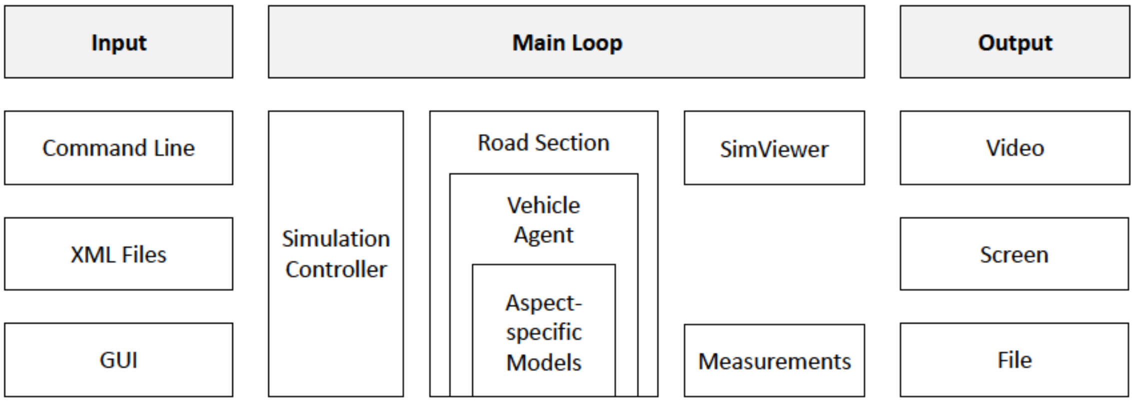

Figure 3.10: MovSim - architecture.

With regard to the modelling capabilities, MovSim can be considered a short-term simulator. Travel decisions refer to the level of agent movements such as acceleration, braking and lane changing. For this purpose, MovSim considers discrete-choice modelling. MovSim does not follow a trip- or activity-based approach for demand modelling as route choices are irrelevant to the agent. Instead, agents represent particles in the network that move in space based on concepts of the applied car-following model. Hence, traffic volume can be defined using numerical input parameters.[2] Given the short-term perspective in modelling movement-related driver decisions, MovSim can be useful to simulate flow behaviour in the field of autonomous driving. The integration of MovSim as a submodule of a larger simulation environment specifically for short-term aspects can be of interest.

ATSim

ATSim (Agent-based Traffic Simulation System) is an application based on the commercial simulator AIMSUN, that extends AIMSUN [154] with agent capabilities. The project was first published in 2011 and has been developed at TU Clausthal [145]. The authors of ATSim argue that for modelling the latest advances in transportation, an agent-based

[2]see `www.traffic-simulation.de` - (access on 14/10/2022)

approach is crucial to represent important aspects of modern transportation such as communication, goals and plans. However, existing agent-based simulators have not focused on an intuitive graphical user interface and exhibit a lack of tools for data collection and data analysis. This is why in the ATSim approach, the commercial simulator AIMSUN has been integrated with the JADE platform [147]. This allows the reuse of all features already implemented in AIMSUN while extending the simulator with agent capabilities. AIMSUN is used for modelling and simulation of traffic scenarios while implementation of agent behaviour is realised in JADE. ATSim has been used to simulate group-oriented traffic coordination in which groups of agents coordinate their speed and lane choices [222]. This can be relevant to simulate vehicle-to-vehicle (V2V) coordination dynamics related to autonomous driving. The ATSim architecture is structured in four components (see Figure 3.11). The first component is the commercial AIMSUM simulator with all its features for modelling and simulating traffic scenarios. The second component is the multi-agent system based on JADE. This component is responsible for managing and controlling the agent life cycle. In ATSim, agents are linked to various types of traffic objects in AIMSUN in order to extend AIMSUN objects with agent capabilities. Communication between agents and traffic objects is possible based on the AIMSUM API. AIMSUM provides an API for the integration of external services in Python and C++. However, JADE is based on Java and it is therefore necessary for ATSim to make use of a middleware in order to allow communication between AIMSUM and JADE.

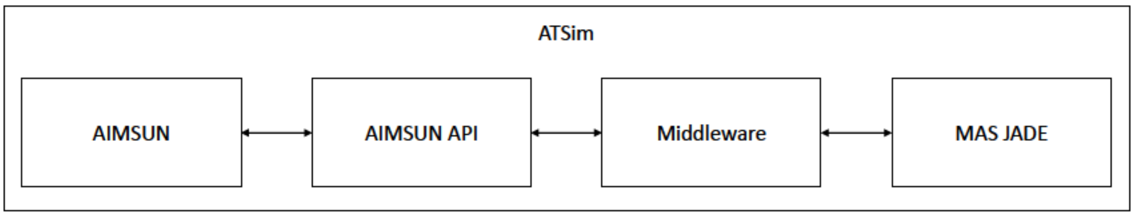

Figure 3.11: ATSim - architecture.

With regard to modelling capabilities, ATSim can be considered a mid- and short-term simulator. Travel decisions refer to the level of route choice but also agent movements based on established car following and lane changing models implemented in AIMSUN. These models have been extended by agent capabilities for modelling the perception and interaction of individual travellers. A distinction is made between *static objects, objects with dynamic states* and *mobile objects*. For example, the road network is represented as a static object whereas traffic lights are modelled as objects with dynamic states and vehicles are presented as mobile objects. Traffic objects can be assigned to an agent in JADE. Each agent can only control a single object in AIMSUM. The link between the agents and traffic objects is based on two assumptions. First, the agent life cycle is synchronised with the life cycle of the associated traffic object. Second, agents constantly receive updated information from the assigned traffic object after each simulation step. AIMSUM follows a trip-based approach for modelling travel demand using origin-destination matrices. The application has been used and thus is suitable for simulating V2V communication and coordination which is of growing relevance with the advancement of autonomous vehicles.

3.4 Discussion

Based on the simulators reviewed (see Table 3.1), it is apparent that the modelling of individuals may vary significantly depending on the area of application as well as the simulated level of detail. In particular, simulation scenarios that are built to study specific research questions require the behaviour of individuals to be modelled for considered time perspectives (short-, mid- and long-term behaviour). The considered time perspective determines which aspects of individual behaviour are included in the modelling (see Figure 3.12). The same also applies to policy-making which needs to address different aspects of individual behaviour depending on the considered problem area. While measures related to road safety primarily aim to achieve change in short-term behaviour e.g. enforcing speed limits in areas that require additional caution, other measures to improve environmental impact or general traffic flow such as electrification of vehicles or promoting the use of shared mobility are intended to induce change in mid- and long-term behaviour. However, finding appropriate measures can be difficult for at least two reasons:

Table 3.1: A summary of reviewed simulators.

Application Name	Area of Application	Licensing	Programming Language	Demand Modelling	Time Perspective on Individual Behaviour
MATSim	Resource Utilisation	GPLv2 or later	Java	activity-based	mid-term
POLARIS	Resource Utilisation	Open-source (license unclear)	C++	activity-based	mid-term
SimMobility	Resource Utilisation	SIMMOBILTITY Version Control License (see Github)	C++	activity-based	long-, mid- and short-term
SUMO + JADE	Connectivity	EPL 2.0 (SUMO) LGPLv2 (JADE) Apache 2.0 (TrasMAPI)	C++, Java	activity-based or trip-based using OD matrices	mid- and short-term
ITSUMO	Connectivity	Open-source (license unclear)	C++, Java	trip-based using OD matrices of LSP	mid-term
MATISSE (DIVAs 4)	Connectivity	GPLv3	Java	trip-based using LSP	mid- and short-term
AgentPolis	New Forms of Mobility	GPLv3	Java	trip-based using OD matrices	mid-term
MovSim	New Forms of Mobility	GPLv3	Java	neither activity- nor trip-based. Only a numeric parameter to specify number of travellers.	short-term
ATSim	New Forms of Mobility	Commercial	C++, Python, Java	trip-based using OD matrices	mid- and short-term

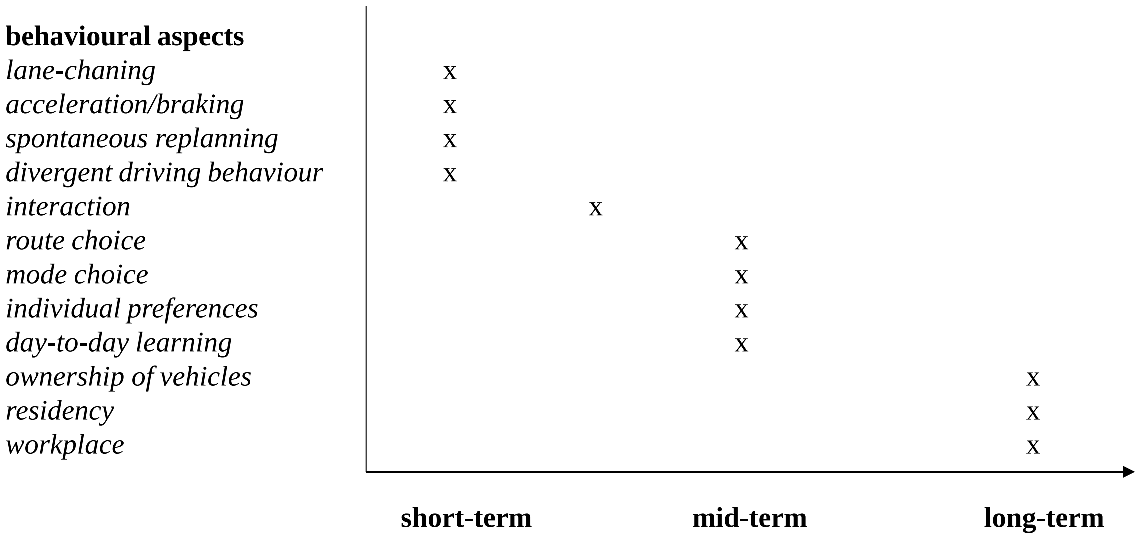

Figure 3.12: Aspects of modelling individuals in traffic simulations.

1. Measures in complex public systems are threatened by rebound effects and therefore change in individual behaviour may not necessarily be in line with the desired outcomes [223]. Car sharing as an example is supposed to encourage people to abandon their private vehicles and thus free up space in urban areas. However, when used by the wrong audiences, car sharing may actually end up worsening the traffic situation. In particular, it has been observed that car sharing services were accepted as an alternative to *public transport*, which in consequence has

increased the number of people travelling in individual vehicles [224].

2. Measures differ in the required time and cost of implementation before showing the desired effects. For example, the expansion or reconstruction of infrastructure is a highly expensive measure that can only be implemented over a longer period of time. The implementation of such a measure typically requires extensive modifications of the spatial territory, which causes considerable damage to the local environment and therefore is often perceived as a highly intrusive intervention. This puts infrastructure projects at risk of running into lengthy debates before even starting with their implementation and thus prevents them from dealing with the original issue at hand. The A49 motorway expansion in Germany [225] is an example of how this type of project is often slowed down due to public opposition.

In order to develop policies that are able to deal with the pressing problems in transportation, it is therefore important to understand the effects of interventions on individuals. This requires simulations to measure the effectiveness and efficiency of policies using performance indicators not only on social benefit but also on the individual level. For example, measuring the happiness of individuals enables the analysis of how policy interventions affect certain groups of individuals and therefore provides information on how these interventions are accepted by the public. This requires simulations to capture more details on the preferences and personal attitudes of individuals. However, there is a lack of concepts to comprehensibly capture preferences and personal objectives as determining factors of individual decisions. Effects on individuals must not be ignored as these are the basic cause of how the system changes under interventions. There are only a few approaches in the field of traffic simulation that even consider the modelling of individual preferences as part of agent behaviour [226]–[228]. [227] have demonstrated how neglecting individual preferences significantly changes simulation results, and have argued that there is a limitation in available traffic simulators for this type of modelling individual behaviour. In order to address this issue, [227] created a simulation model that uses individual preferences as part of utility functions for agent decisions. Preferences are based on survey data and vary depending on the three agent attributes: *gender, age* and *income*. While this type of preference-based utility function is a common practice in economics to model quantitative decision-making, this model does not consider that the perception of preferences changes depending on the context of the simulated activity. For example, *time/punctuality* has a different value when commuting to work as compared to a social visit. Hence, travel behaviour is specific to the context of travel (travel purpose). The ASIMUT approach [228] has recognised this problem and therefore introduces a weighted sum model on preferences that is used for calibration. As travel decisions in the real world are driven by travel purposes, individuals need to be modelled to have knowledge not only about traffic but also about the purpose of travel. Thus, there is a need for appropriate structures to flexibly model knowledge as well as individual preferences as determining factors of purpose-driven travel behaviour. The detailed modelling of individuals can be very time-consuming and usually requires a deep understanding of the underlying system architecture. It is therefore important to ensure the reusability of modelled individuals and to develop appropriate customisation options to keep the modelling overhead to a minimum.

To conclude this review, the main challenges in the modelling of individuals in traffic simulations can be summarised as follows. To address these challenges, this thesis further looks into improving the modelling of individuals in Chapter 4.

1. There is a lack of concepts to capture preferences and objectives as determining factors of agent decisions which can be particularly relevant to measure the effects of policy interventions not only on social benefit but also on individuals.

2. There is a need for appropriate methods to capture the decision-making of individuals that consider not only traffic-related aspects but also the simulated activity.

3. There is a need for appropriate methods that are able to minimise the complexity during the modelling/ customisation process and to ensure the reusability of agents when modelling the details of individuals.

3.5 Summary of Chapter 3

In this chapter, a systematic survey was performed to give an overview of the capabilities of available traffic simulators to model individuals and their behaviour based on current topics in mobility. Section 3.1 illustrates the methodology for the systematic survey. Based on this, the survey yielded a large number of simulators that vary in their use of multi-agent technology as well as their scope of application. In section 3.2, simulators have been categorised and aspects for comparison have been defined for a more detailed review. As the focus of this survey was on modelling individuals, particular interest was given to simulators that were either fully agent-based or used agent technology for certain aspects of the simulation (hybrid). Available literature showed that current topics in mobility mostly revolve around the three application areas: 1. resource utilisation, 2. digital connectivity and 3. new forms of mobility. Simulators in this survey have been grouped according to their area of application and for each application, three examples have been examined in detail based on their previous use cases. In section 3.3, these examples have been reviewed with regard to the implemented features for modelling aspects of individual behaviour. The discussion in section 3.4 has found that aspects of individual behaviour differ significantly depending on the simulated time perspective. For example, long-term scenarios such as studies of land use look at aspects such as decisions of individuals on home and work locations, while mid- and short-term scenarios model aspects such as mode choice or lane changing and braking behaviour. Related work showed that the options for modelling knowledge and preferences of individuals as determining factors of agent decisions are limited. However, this can be particularly relevant to measure the effects of policy interventions on individuals. Furthermore, travel decisions of individuals need to be linked to the simulated activity. As modelling these details increases the complexity of simulations, there is a need for innovative modelling methods.

Chapter 4

Modelling Individual Preferences to Study and Predict Effects of Traffic Policies

With the application of agent-based models, traffic can be simulated as a complex adaptive system in which systemic patterns arise as emergent phenomena. System behaviour is the result of interactions between personal behaviours and preferences of a large set of individuals. Assuming rational behaviour such systems typically move towards some form of homeostasis (equilibrium). Policy-making can help guide individuals towards a socially beneficial homeostatic solution but requires a deep understanding of the individual. Hence, modelling of individuals is a crucial part of building simulation models to study the effects of policies. Available traffic simulation models lack concepts to comprehensibly capture preferences and personal objectives as determining factors of individual decisions (see Chapter 3.4). Modelling these details accounts for extra effort and therefore requires appropriate modelling structures. The application of semantic technology that typically focuses on enhancing the meaning and context of information can help address these issues. In particular, if knowledge is formulated in a form ensuring that information can be easily managed, additional knowledge can be inferred through the application of reasoning mechanisms. Meanwhile, the use of semantic technology may potentially increase the complexity of the overall architecture of the system as well as the simulation model but allows agent knowledge and behaviour to strictly be separated from the implementation of the simulation. As a result, agents can be flexibly reused across different domains and in varying scenarios, requiring only marginal changes in the concrete implementation. The application of semantic technology thus facilitates modifications or extension of agent knowledge about their purpose of travel when changing agent activities for simulating different scenarios. For example, agents from a commuting scenario that have been modelled with knowledge about basic concepts in traffic (e.g. mode options) can be transferred to other scenarios that simulate the mobility of individuals during their grocery shopping (knowledge about basic traffic concepts can be reused while agents can be extended with knowledge on the grocery shopping domain).

This chapter focuses on creating a simulation model that is able to capture knowledge and preferences as determining factors of individual decisions. This is a prerequisite for examining how policy-making in mobility affects both individual as well as global system behaviour. Note that the focus of this work is not to present a validated simulation model but to demonstrate how the proposed methods can be used to model and simulate what-if scenarios to measure the effects of policy interventions in the simulations. For this purpose, the modelled scenarios demonstrate how effects of traffic policies on individuals can be investigated. The experiments will focus on showing how the simulations can respond appropriately by producing plausible changes in behaviour for different input settings. Based on these objectives, the following section provides a brief recap on applying semantic technology to agent-based simulations. Following this, a detailed description is given of the proposed method for modelling preferences and knowledge of individual traveller agents. As an example, simulation is performed for a scenario that deals with mobility caused by individuals during their grocery shopping. Furthermore, a simulation of a leisure trip scenario is presented to demonstrate how agents that are modelled based on the proposed method can be reused across different domains. The chapter closes with a discussion of experimental results as well as the strengths and limitations of the proposed method. Earlier versions of parts of this chapter have been published in [1], [3], [6], [7].

4.1 Semantic Modelling

Agent decisions in traffic are influenced by numerous aspects and therefore require a broad knowledge of the world. For this purpose, ontologies can be used. The term ontology refers to the science of describing entities in the world and originates in philosophy where it is used with a more general meaning. Ontologies are an expressive tool to model domain knowledge in a machine-readable form and provide an explicit, shared specification of a conceptualisation of that domain[229]. Ontologies typically consist of a taxonomy of concepts each with properties and relations which allows for representation of semantics. They have become a common instrument for consistently documenting knowledge that can be shared among machines. As [230] formulates, formally, an ontology O is a triple $O = (C, R, I)$ with C a set of concepts, R a set of relations, and I a set of individual objects. Concepts represent classes that define sets of objects. An object that belongs to a concept is called an instance of that concept. Objects can be linked through relations $r \in R$ which are also called object properties. These relations are defined with concepts as their domain and range, and subsequently, instances instantiate these relations. Therefore, r is a set of pairs (c, d) with $c, d \in I$. Instances $i \in I$ can have data properties that have primitive data (e.g. numbers or strings) as the range. Hence, data properties are sets of pairs (i, p) with p a primitive value. The *Web Ontology Language (OWL)* is a formal notation standard to define and create ontologies [231]. It is part of the *Semantic Web Activity* which describes the idea of computers with structured collections of information and sets of inference rules that are able to perform computational reasoning [232]. OWL is based on description logic and decidable subsets can be defined that facilitate the application of reasoning algorithms. The combination of *OWL* together with *Semantic Web Rule Language (SWRL)* [233] is a frequently used set of tools. OWL can be used to create a structured collection of information while SWRL, as an extension, is a W3C proposed language to express *If-Then* implications based on Horn Logic (see [234]). The application of semantic technology to extend agents with semantically modelled knowledge bases has been demonstrated for market simulations [235].

4.2 Method

This work proposes an abstract model for representing the knowledge and preferences of individuals. At the center of this method is a semantic framework that extends traditional agent programming with qualitative modelling of decision aspects. Based on this, it can be particularly useful to separate general agent activity logic from aspects of modelling agent knowledge. While basic operations of the agents remain part of traditional agent programming, agent knowledge can be shifted into separate ontologies (see [229]) which can be used for creating knowledge bases. [235] have demonstrated the effectiveness and efficiency of this approach. In addition to this, the separation of agent knowledge from its operating behaviour makes it possible to look into the field of knowledge engineering to structure agent knowledge. In particular, models on generalisation and abstraction of information can help to reduce complexity in the modelling and improve the reusability of modelled agent knowledge.

An example of this is the *CommonKADS* model which proposes a three-layered architecture for structuring different types of knowledge [236]. The lowest layer *domain knowledge* defines relevant concepts as well as simple relations for a modelled subject area. Concepts from domain knowledge are put into a logical context which enables derivation of *inference knowledge* in the second layer. Based on this, information from the lower layers is used for determining action strategies which in the top layer is referred to as *task knowledge*. The following example shows how the different layers of knowledge can be reflected in a real-world scenario (see Figure 4.1): Given a scenario in which a mechanic repairs bicycles, domain knowledge typically contains information about the components of a bicycle, potential technical problems that may occur, as well as possible causes and corrective measures. Inference knowledge puts this information into a logical context and hence contains knowledge about the assembly or mechanics of bicycles. Task knowledge combines information from the lower layers allowing decision-making on appropriate actions, e.g. diagnosis of technical malfunctions of a bicycle and procedures to repair the damage.

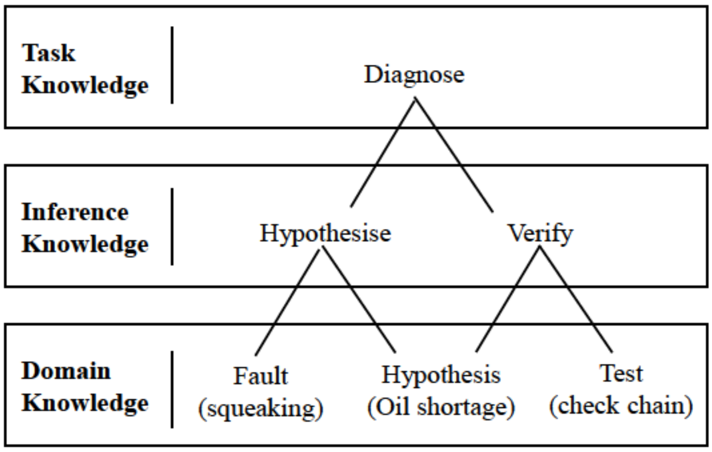

Figure 4.1: An example of the CommonKADS knowledge structure.

Here an agent architecture is proposed that models preferences and knowledge of travelling individuals analogous to the structures of CommonKADS (see Figure 4.2). Concepts and relations in the layers for domain and inference knowledge are modelled as OWL ontologies [231]. In particular, domain knowledge can be split into separate ontologies that differ in the scope of the modelled subject area. Each ontology is part of the information that is available to the agent. The combination of multiple ontologies that each focus on different aspects of the scenario produces a comprehensive model of world knowledge.

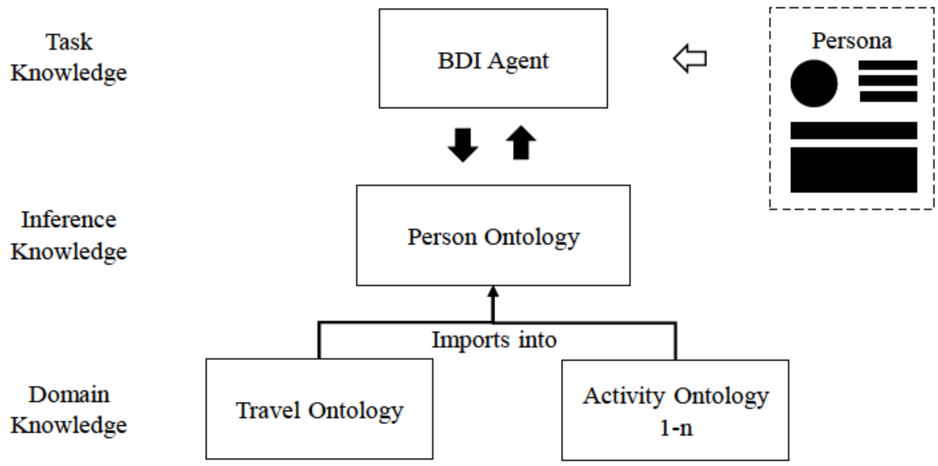

Figure 4.2: Agent knowledge architecture.

Traffic simulations typically share the same basic concepts on infrastructure (e.g. roads, road signs, traffic lights) or modes of travel (car, bus, bicycle etc.). Hence, domain knowledge should include a dedicated *travel ontology* that exclusively captures common traffic domain concepts. This will allow instances of this ontology to be re-used across different traffic scenarios. For example, the same travel ontology is likely suitable for modelling both commuting within a city as well as modelling leisure trips. Both scenarios differ in the activity performed which is why agents require their own activity-specific domain knowledge. Domain knowledge therefore needs to be extended with a second type of ontology (*activity ontologies*) to capture relevant knowledge about the purpose of travel. For example, an activity ontology on commuting typically provides agents with information about places of work, while an activity ontology on leisure trips would provide agents with information about landmarks, shops, or sports centers. Agents can be equipped with different ontologies that each can be "plugged in" and extended or replaced as needed to address different research questions. For more complex scenarios (e.g. when simulating agents that perform a series of different activities throughout the day), agents can also be equipped with multiple activity ontologies. In order to combine information from the travel and activity ontologies, the agent architecture includes an additional central *person ontology* which is a representation of the collective knowledge that is available to the agent. In addition to domain knowledge, this ontology also comprises information about person-specific concepts such as census properties (age, gender, occupation, etc.) which are prerequisites for inferring further agent attributes such as individual preferences.

Inference knowledge is represented in SWRL rules [233] which extend OWL to express *If-Then* implications. The rules are used to define how observed real-world information is transferred into the attributes of the agents. In particular, individual preferences defined in the travel and activity domains are inferred from survey data. Census properties are given to the agent by assigning persona profiles according to the local population (see [7]). These properties are reflected in the ontology and serve as input to rules to infer preferences. Persona profiles in this example are derived from the results of a classification of data from a consumer study [237] which represents the most relevant groups of individuals in the German demographic (see Figure 4.3). They are categorised by current life stage, *family status* as well as *social strata/income* (as illustrated in [238]). Inferred information in the ontology can be accessed by the agent when implementing its decision behaviour in the layer dealing with task knowledge. Implementation of agent decision behaviour is based on the standard BDI agent architecture which allows agents to perform the selection of actions (intentions) on the basis of their goals (desires) and preferences (beliefs), given their travel and activity domain knowledge (beliefs) [101]. For demonstrating the proposed methods, the theoretic concepts have been implemented in a new simulation environment that is called AGADE Traffic simulator [239]. This new simulator provides BDI action selection and definition of agent desires using JADEX [240] while beliefs are expressed with OWL language constructs augmented with SWRL rules. Note that the proposed model uses semantic technology with its inference mechanisms throughout the implementation of agent behaviour.

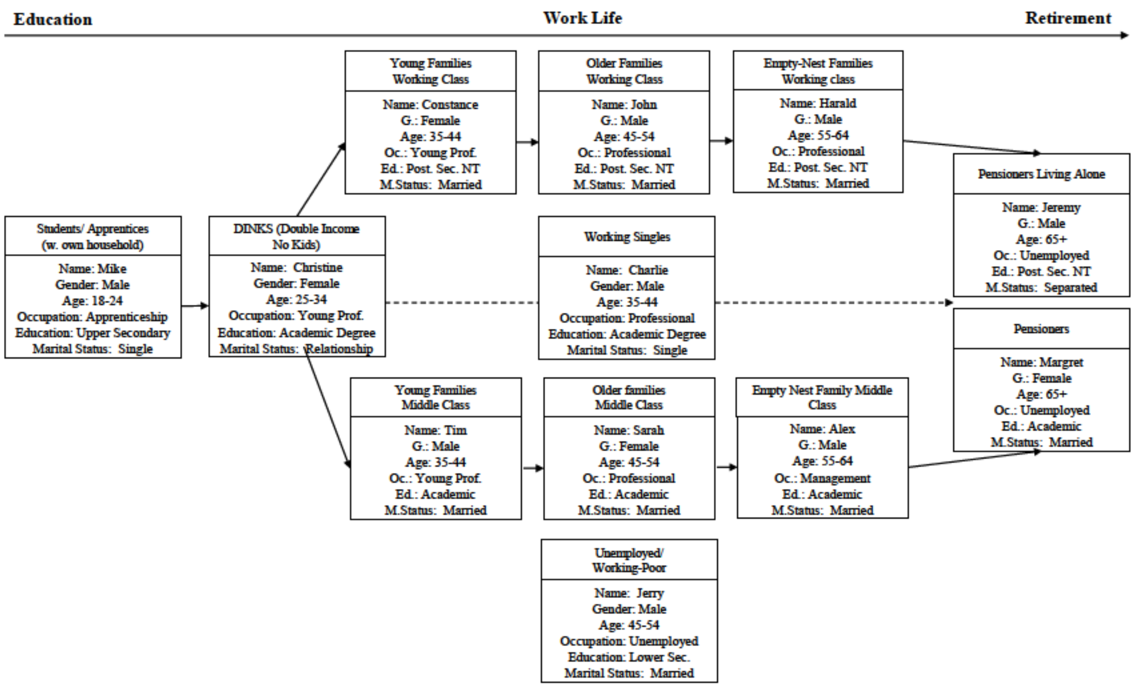

Figure 4.3: Persona profiles based on social census.

4.3 Example

As demonstrated in chapter 3, current research on traffic has primarily studied the effects of policies by measuring social benefit and thus not sufficiently considered effects on individuals. However, effects on individuals are the basic cause of how the system changes under interventions which is why they cannot be ignored. In order to evaluate effects on individuals, simulation results need to include information on individual utility which requires a more detailed modelling of individuals that is able to capture preferences as determining factors of agent decisions. Furthermore, decisions and preferences vary depending on the context of travel, which is why agents require knowledge about the simulated activity. The next sections demonstrate how the proposed method addresses these issues by applying them to two mobility scenarios that differ in the simulated purpose of travel (activity).

4.3.1 Mobility of Individuals during the Activity of Grocery Shopping

The following scenario simulates the mobility of individuals that is associated with their grocery shopping. Agents are assigned a randomly generated list of food items selected from a set of products available in the supermarkets of the simulation. This set is categorised (e.g. fruit, vegetable, grains) and probability distributions over the categories can be defined and assigned to different agent types. Agents aim at purchasing items in their lists in the course of which they have to make decisions, e.g. choosing a supermarket together with a mode of travel.

To illustrate how the framework can be applied, the following grocery shopping scenario has been defined. The scenario contains a set of supermarkets that differ not only in location and product availability but also in the attributes of products such as their price and whether they are organic. This type of information is modelled in an activity ontology for the *food/grocery shopping* domain. Agents are randomly assigned a list of food items from this ontology to purchase. In particular, items on this list are categorised (e.g. fruit, vegetable, grains) and probability distributions over the categories can be defined and assigned to different agent types (persona profiles). In order to acquire all items on their shopping list, agents may have to visit multiple supermarkets. For the purpose of this example, it is assumed that agents are generally willing to visit more than one supermarket rather than not acquiring the items on the shopping list. In addition to this, the travel domain ontology in this scenario comprises information about possible traffic mode options (walking, bicycle, car, public transport) that agents can pick to get to the supermarkets. During their grocery shopping activity, agents have to determine a combination of supermarkets and appropriate modes of travel. This type of decision behaviour is implemented in the layer of task knowledge which uses information from domain and inference knowledge. In particular, inference knowledge contains information about individual preferences that are derived from rules based on survey data. Implementing the process to derive the preferences using rules in the ontology allows scenario-specific preference concepts to be modelled as part of domain knowledge for the simulated activity. Furthermore, in the layer of inferences knowledge, preferences become instantiated by applying rules that use survey data to compute the actual values of preferences for an agent. The grocery shopping scenario demonstrates how travel decisions such as mode selection as well as determining the travel destination depend not only on information from the traffic domain but also on information from the simulated activity which in this case is the supermarket/food domain. For example, in addition to knowledge about basic traffic concepts such as available mode options, agent decisions also require scenario-specific knowledge about the type of food on the shopping list as well as at which type of supermarket these items can be purchased. Preferences and domain knowledge are then used as input for agent decisions that are based on utility functions and algorithms. Based on this, the agent life cycle can be defined by the following three phases (see Figure 4.4).

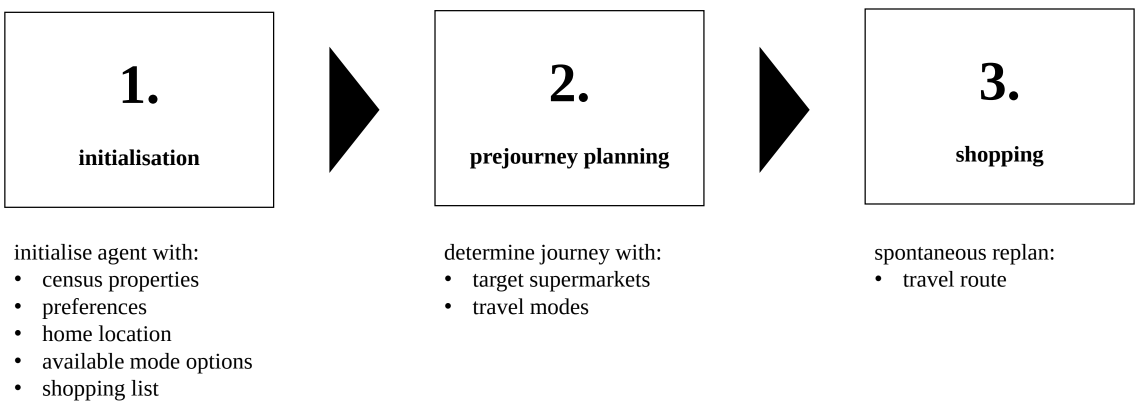

Figure 4.4: Agent life cycle.

An agent a has a set of attributes A. A is the disjoint union of descriptive attributes Δ, and preference attributes $\Phi = T \cup F$ with traffic related preferences T and food related preferences F. While ranges of attributes in Δ are all nominally scaled, attributes in Φ take values from a *Likert scale* of 1 to 5 (1="not important" and 5="very important"). The selection of attributes relevant for modelling is based on behavioural surveys on mobility [241], [242] and grocery shopping [243] (see Table 4.1).

The agent population conforms to the principal structure of the group of persons from which survey data was collected. Survey data are used to create agents so that the empirical frequency distribution $D_{persona}$ is preserved in the agent population. For each agent, values for the attributes in Δ are determined by their persona. Now, values for preferences $\phi \in \Phi$ are derived depending on the values of the attributes in Δ, again according to the survey data. For this purpose, rules have been modelled in the ontology with which a specific categorical probability distribution over the Likert scale values can be derived for each preference $\phi \in \Phi$. Therefore, for each preference $\phi \in \Phi$, and for each nominal attribute $\delta \in \Delta$, there is a conditional probability distribution $D_\phi(L \mid \delta)$ over the values of the Likert scale $L = \{1,2,3,4,5\}$ depending on the value of δ. The probabilities $p_\phi(L \mid \delta)$ are determined by the empirical frequencies in the surveys. It is assumed that the preferences of an agent are influenced by the entire set of its descriptive attributes $\delta \in \Delta$ the corresponding probabilities are aggregated over Δ into the weighted sum $p_\phi(l) = \sum_{\delta\in\Delta} \lambda_\delta \cdot p_\phi(l \mid \delta)$ with $\sum_{\delta\in\Delta} \lambda_\delta = 1$ giving a probability distribution for each $\phi \in \Phi$. In this work, all attributes are weighted as of equal importance, i.e. $\lambda_\delta = \frac{1}{|\Delta|}$.

An example will illustrate this: Let $\Delta = \{age, occupation\}$ be the set of descriptive attributes and Φ consist of a single preference *Environmental Impact* lying in T meaning

Table 4.1: Attributes and preferences assigned at initialisation of an agent.

Descriptive attributes Δ	**Traffic preferences T**	**Food preferences F**
age	flexibility	price tendency
education	time	product quality
gender	reliability	eco-friendliness
occupation	privacy	fair trade
marital status	safety	
	monetary costs	
	environmental impact	
	convenience	

that there are no food preferences in F. Furthermore, let a_1 with $\Delta_{a_1} = \{$*18-25, student*$\}$ and a_2 with $\Delta_{a_2} = \{$*46-55, factory worker*$\}$ be agents. Given the specific values of their descriptive attributes, a_1 and a_2 are expected to have different preferences manifested in the values for $\tau \in T$, i.e. their awareness for *Environmental Impact*. Indeed, the survey data gives evidence that persons of $age = 18\text{-}25$ show a higher awareness of environmental issues. This is modelled in $p_{EnvironmentalImpact}$ (see Table 4.2). For the second descriptive attribute *occupation* again probabilities for $l \in L$ are drawn from the empirical distribution of data in the survey (see Table 4.2). The weighted sum of the values for *age* and *occupation* yields $p_{\mathit{Environmental\ Impact}}(l)$ for each $l \in L$. Roulette wheel selection is then used, based on the aggregated probabilities $p_\tau(l)$, to determine the value l which is then assigned to preference τ. This concludes the initialisation phase of agent a. Computation of the probabilities $p_\tau(l \mid \delta)$ uses rules of the ontology of the following scheme which allows information from empirical data to be linked to domain knowledge:

$$Agent(?a) \land Preference(?\tau) \land hasCensusProp(?a,?cprop) \land swrlb:equal(?cprop,\delta) \Rightarrow$$
$$Agent(?a) \land hasPreference(?a,?\tau) \land Preference(?\tau) \land hasLikert1(?\tau, p_\tau(1 \mid \delta)) \land hasLikert2(?\tau, p_\tau(2 \mid \delta)) \land$$
$$hasLikert3(?\tau, p_\tau(3 \mid \delta)) \land hasLikert4(?\tau, p_\tau(4 \mid \delta)) \land hasLikert5(?\tau, p_\tau(5 \mid \delta))$$

Based on this, the following SWRL rule can be defined to determine probabilities $p_{\mathit{Environmental\ Impact}}(l \mid \mathit{age} = \text{"}18-24\text{"})$ for the example agent $a1$:

$$Agent(a1) \land Preference(EnvironmentalImpact) \land hasAge(a1,?age) \land swrlb:equal(?age,\text{"}18-24\text{"}) \Rightarrow$$
$$Agent(a1) \land hasPreference(a1,?EnvironmentalImpact) \land Preference(EnvironmentalImpact) \land$$
$$hasLikert1(EnvironmentalImpact, p_\tau(1 \mid 0.05)) \land hasLikert2(EnvironmentalImpact, p_\tau(2 \mid 0.1)) \land$$
$$hasLikert3(EnvironmentalImpact, p_\tau(3 \mid 0.15)) \land hasLikert4(EnvironmentalImpact, p_\tau(4 \mid 0.3)) \land$$
$$hasLikert5(EnvironmentalImpact, p_\tau(5 \mid 0.4))$$

Table 4.2: Example for preference probabilities for agent a_1.

Probabilites/Likert Values l	**1**	**2**	**3**	**4**	**5**
$p_{\mathit{Environmental\ Impact}}(l \mid \mathit{age} = \text{"}18-24\text{"})$	0.05	0.1	0.15	0.3	0.4
$p_{\mathit{Environmental\ Impact}}(l \mid \mathit{occupation} = \text{"}student\text{"})$	0.1	0.1	0.2	0.3	0.3
$p_{\mathit{Environmental\ Impact}}(l)$	**0.075**	**0.1**	**0.175**	**0.3**	**0.35**

In the second phase of the agent life cycle (prejourney planning) the agent makes decisions about supermarkets to be visited as well as appropriate modes of travel according to its personal preferences. For this purpose, preferences of the agent are used as input arguments for compound utility functions defined below (see Equation 4.1 and 4.2). Based on this, the agent successively constructs a shopping journey consisting of legs from supermarket to supermarket (and from home to the first supermarket and back home from the last) with appropriate travel modes. Supermarkets and modes of travel are chosen to maximise the utility of the agent. Note that these decisions are mutually interdependent and have to happen simultaneously e.g. distant supermarkets can only be reached by car while choosing to walk will likely determine the agent to choose a nearby supermarket. Thus, decision-making is multi-criterial as agent behaviour is not only determined by traffic-related aspects but also by individual preferences relevant for the selection and purchasing of food items.

In particular, the utility has been defined that reflects traffic-related preferences of an agent a. For a given attribute $\tau \in T$ (T the set of traffic-related attributes) and a traffic mode $m \in M$ (M the set of available traffic modes), let $u(\tau, m)$ be the given utility of mode m with regard to a specific mode attribute τ and a_τ the preference value of τ for agent a. Spontaneous modal change during the journey accounts for extra effort and therefore involves costs which are modelled with a symmetric function $c : M \times M \to \mathbb{R}$ with $c(m, m')$ the associated cost for changing from mode m to mode m' with $c(m, m') = 0$ for $m = m'$. Note that an artificial mode m_{null} has been added to represent the start of the food shopping journey and that $c(m_{null}, m) = 0$ for all $m \in M$. Based on this, the total traffic utility U_{TT} of traffic mode m for agent a is defined. Note that the value of this function also depends on the traffic mode m_c of the last leg.

$$U_{TT}(a, m, m_c) = \sum_{\tau \in T} u(\tau, m) \cdot a_\tau - c(m_c, m) \tag{4.1}$$

Supermarkets $s \in S$ (S the set of supermarkets) are assigned utilities $u(f, s)$ that rate their products with regard to $f \in F$ (F the set of food-related attributes) (see Table 4.1). Furthermore, a_f is the value for preference f of agent a. Based on this a shopping utility $U_F(a, s)$ is determined:

$$U_F(a, s) = \sum_{f \in F} u(f, s) \cdot a_f \tag{4.2}$$

Furthermore, supermarkets are assessed by the degree to which the products they stock cover the items on the shopping list of an agent as well as their distance to the current location. If agent a has r_a open items on its list, q_s of which are available in supermarket s, then the quotient $\frac{r_a}{q_s}$ quantifies the product coverage of s for a. Furthermore, for each agent a a randomly generated value e_a models aversion of a towards additional trips to other supermarkets based on probabilities provided by [244]. The *euclidean distance* $d(a,s)$ from the current position of a to the supermarket s is used as an estimate for the travel distance to s as the agent at this stage does not know the real travel distance which is determined later by the actual route. For each agent values for U_{TT}, U_F and $d(a,s)$ are normalised with *min-max normalisation* so that they lie in [0,1]. As decisions on the mode of travel and selection of supermarkets are interdependent, traffic and food related utilities are aggregated into a single utility function with which an agent determines the next supermarket to go to and how to get there. Therefore, the leg $r = (m,s)$ to the next supermarket s is an element in $M \times S$ (with M travel modes and supermarkets S) that has a utility:

$$U(a,r,m_c) = (1 - d(a,s)) + U_{TT}(a,m,m_c) + U_F(a,s) + \frac{r_a}{q_s} * e_a. \tag{4.3}$$

Algorithm 1 shows how an agent successively selects supermarkets and determines rides that are concatenated into a journey. It is assumed that the overall supply of all supermarkets covers all items on shopping lists and that items are abundantly available. Note, that no additional optimisation is performed with respect to the order in which the supermarkets are visited, as the intention is to simulate the natural behaviour of individuals. This concludes the prejourney planning phase for agent a.

Algorithm 1 Algorithm to determine agent journey.

Require: agent a, list of shopping items I_a, supermarkets S, traffic modes M

journey= empty list;
while I_a is not empty **do**
    $r = (m,s) = \underset{r \in M \times S}{\operatorname{argmax}}\, U(a,r)$
    journey=journey+r
    I_a=I_a\supply(s)
end while
return journey

Finally, the agent enters into the *shopping phase* of the simulation. Travel decisions from prejourney planning are primarily based on preferences derived from survey data, which do not change. Hence, target supermarkets as well as selected modes of travel remain unchanged during the shopping tour. However, agents may spontaneously change routes depending on the current traffic load. Routing uses the A* algorithm based on shortest time [245]. Let W be a route with $w \in W$ being a continuous section of route W with the same speed limit $v(w)$. The travel speed of an agent is defined $v(w,m) = min\{v(w), v(m)\}$ for $v(m)$ the maximum speed of travel mode m. Furthermore, $d(w)$ defines the distance to be covered on w and $n(w)$ an indicator for the present traffic load. Thus, overall travel time T is computed:

$$T(W,m) = \sum_{w \in W} \frac{d(w)}{v(w,m)} + n(w) \tag{4.4}$$

The shopping phase terminates once all scheduled supermarkets in the journey have been visited and the agent has returned to its home location.

As an example, a simulation has been created for the German city of Wetzlar. Wetzlar counts a total of 29 supermarkets based on data provided by Google Maps. According to the German census of 2011 [246], the population in Wetzlar consists of approximately 50,000 citizens that are spread over 20 residential areas. It is assumed that one person shops for their entire household and that 20% of these households shop during the simulated time interval. Thus, a population of 2130 agents has been generated that replicates the empirical distribution of residents. Agents in the population are assigned a persona profile (as illustrated in Figure 4.3) which defines values for their descriptive attributes Δ. Based on this, agent preferences are computed according to the descriptions provided above. Note that in this work the implementation uses stochastic elements only while computing preference values, thus keeping the subsequent decision processes deterministic, and in consequence not requiring multiple runs of the simulations. This simplifies analysis and proof of concept making comparison of simulations easier.

For experimentation, three simulation runs (A, B and C) have been performed with identical agent populations in order to examine how different interventions have an effect on traffic. Simulation A reflects traffic without interventions and therefore serves as the benchmark scenario. In comparison to this, simulation B assumes as an intervention an educational campaign that achieves a change of attitude for 35% of the inhabitants (756 agents) to traffic and its environmental consequences. This is modelled by increasing traffic preferences on *Environmental Impact* and reducing traffic preferences for *Convenience*. In particular, 42 of these agents changed their preference value on *Environmental Impact* from 1 to 5, 200 from 2 to 5, and 514 from 3 to 5 while at the same time 150 of these agents changed their preferences on *Convenience* from 5 to 1, 295 from 4 to 1, 198 from 3 to 1 and 101 from 2 to 1. For simulations A and B, agents are modelled to have access to all modes of travel. However for simulation C, as an intervention ownership of private vehicles has been limited for 38.5% of the population (821 agents), forcing these agents to switch to alternative mode options. It is assumed that utilities $u(\tau, m)$ for mode options remain the same across all simulations A, B and C (see table 4.3).

Table 4.3: Mode utilities on a scale of 0 to 8 for simulations A, B and C

	Car	**Bike**	**Walking**
Flexibility	8	2	1
Time	8	3	0
Reliability	7	7	8
Privacy	8	1	0
Safety	7	2	5
Environmental Impact	0	8	8
Monetary Costs	2	6	8
Convenience	8	0	0
$\sum$	**48**	**29**	**30**

Given that agents in simulation A have access to all modes of travel and that mode utilities from Table 4.3 generally favour car usage, simulation results in A show that most of the agents have chosen to travel by car (see Table 4.4). It can be assumed that policymakers prefer agents to choose green transportation modes such as *walking* or *cycling* to avoid the emission of exhaust gases. In the simulation, this is mirrored through key performance indicators on aggregated travelled distances. Environmental impact is measured by the indicators *global travel distance* which is the sum of the overall distances travelled by the set of all agents, and *combustion distance* that only

Table 4.4: Comparison of modal choices for simulations A, B and C

	A	**B**	**C**
Car	99.95%	99.72% (⇓ 0.23%)	61.41% (⇓ 38.5%)
Bike	0.05%	0.14% (⇑ 0.09%)	37.98% (⇑ 37.93%)
Walking	0.00%	0.14% (⇑ 0.14%)	0.61% (⇑ 0.61%)

Table 4.5: Indicators on environmental impact for simulations A, B and C

	A	**B**	**C**
Global Travel Distance [km]	11453	11393 (⇓ 0.53%)	10213 (⇓ 10.82%)
Combustion Distance [km]	11452	11376 (⇓ 0.66%)	7115 (⇓ 37.87%)

considers modes of travel that produce exhaust gases (see Table 4.5). Thus, in this work, the focus is on examining to what extent interventions from simulations *B* and *C* are suitable to achieve the desired change.

The educational campaign from simulation *B* produces only a minimal shift in modal choices. In particular, car usage has been reduced (⇓ 0.23%) while the number of cyclists (⇑ 0.09%) and pedestrians (⇑ 0.14%) has increased. In principle, the observed course of change is favourable, but given that only a very small percentage of agents exhibit a change in behaviour, desired effects on global indicators are hardly noticeable with *global travel distance* decreasing by ⇓ 0.53% and *combustion distance* decreasing by ⇓ 0.66%. In comparison to this, the intervention from simulation *C* achieved a more significant effect. Agents that were denied ownership of a private car were forced to switch to alternative mode options and in consequence the number of cyclists (⇑ 37.93%) and pedestrians (⇑ 0.61%) increased. Performance indicators on environmental impact also show a decrease on *global travel distance* (⇓ 10.82%) and *combustion distance* (⇓ 37.87%). Hence, from a global perspective, a ban on vehicles is more effective than trying to achieve a change of attitude. However, real-world implementation of such a measure has strong effects on individuals which causes public opposition. As the proposed model includes more details in the modelling of individuals, it is possible to measure exactly these types of effects on individuals in the simulations. For assessing interventions in a system by (individual) utility, it is necessary to take a utilitarian perspective on utility [247]. Utility as experienced by individuals has been associated with happiness measures [248]. Following this idea, utility functions can be used to quantify experienced utility as an indicator for the satisfaction of individuals. It can be noted that this relation between utility and happiness is debatable, but so far there is no consensus on this matter (see [247] for a discussion). This analysis specifically looks at the utility of agents affected by the intervention (see Table 4.6). With regard to simulation B, a total of 756 agents are affected by the educational campaign. The individual utility of these agents in simulation A averages 0.60642 in comparison to a utility value of 0.58036 in simulation B. This indicates a decrease ($\Downarrow$ 4.29%) in the experienced utility of affected agents in simulation B. In contrast, utility values of agents affected by the vehicle ban in simulation A averages 0.59892 which in simulation C decreases by $\Downarrow$ 12.26% to a value of 0.5255. This shows that the effects of interventions on individuals for the vehicle ban (C) are more intrusive in comparison to the educational campaign (B), which increases the risk of public resistance.

Table 4.6: Normalised average traveller utility of changed agents for simulations A, B and C

	A	**B**	**C**
Educational campaign	0.60642	0.58036 ($\Downarrow$ 4.29%)	-
Vehicle ban	0.59892	-	0.5255 ($\Downarrow$ 12.26%)

As mode utility in all three simulations A, B and C significantly favours car usage, utility values of cycling and walking have been recalibrated to reduce the utility gap between mode options (see Table 4.7). In the real world, this models an improvement in the quality of alternative travel modes, e.g. through measures that generally facilitate travel conditions for pedestrians and cyclists, and thus reduce the advantage of cars. Simulations have then been rerun with the same populations and configurations from A, B and C, to look at how the effects of interventions change under new circumstances. Simulations from this iteration of experiments are referred to as $A2$, $B2$ and $C2$.

Although car users still account for the majority of travellers in this iteration of experiments, modal choices in the new benchmark simulation $A2$ (see Table 4.8) show a greater representation of cyclists and pedestrians in comparison to the observed modal split in the original simulation A. This time, the effects of the educational campaign in simulation $B2$ are more pronounced showing a shift from car users to

Table 4.7: Mode utilities on a scale of 0 to 8 for simulations A2, B2 and C2

	Car	**Bike**	**Walking**
Flexibility	8	6 ($\Uparrow$ 4)	4 ($\Uparrow$ 3)
Time	8	4 ($\Uparrow$ 1)	1 ($\Uparrow$ 1)
Reliability	7	7	8
Privacy	8	6 ($\Uparrow$ 5)	4 ($\Uparrow$ 4)
Safety	7	2	6 ($\Uparrow$ 1)
Environmental Impact	0	8	8
Monetary Costs	2	6	8
Convenience	8	1 ($\Uparrow$ 1)	1 ($\Uparrow$ 1)
$\sum$	**48**	**40** ($\Uparrow$ 11)	**40** ($\Uparrow$ 10)

cyclists by 34.65%. This development is also reflected in the performance indicators on environmental impact (see Table 4.9). While *global travel distance* only decreases by $\Downarrow$ 3.03%, *combustion distance* is reduced by a total of $\Downarrow$ 29.77%. In contrast, limiting the ownership of vehicles in simulation $C2$ achieves a similar outcome as in $B2$. Results of $C2$ also show a shift from car users to cyclists by 32.72% which in consequence reduces *global travel distance* ($\Downarrow$ 6.57%) as well as *combustion distance* ($\Downarrow$ 34.96%). The comparison of results from $B2$ and $C2$ with output data from the original simulations B and C suggests that measures can yet become more effective when the quality of alternative mode options increases, and thus the advantage in utilities of the car is reduced. Furthermore, utility values in Table 4.10 show that the supposedly harsh intervention of the vehicle ban in $C2$ is perceived as less intrusive with individual utility decreasing by only $\Downarrow$ 2.88% in comparison to the original simulation C ($\Downarrow$ 12.26%) when there are genuine mode alternatives to the car option. Overall, simulation $C2$ achieves the best outcome with regard to indicators on environmental impact but $B2$ is almost equally effective while using a less intrusive intervention (educational campaign) which reduces the risk of public opposition. This concludes simulation of the first example scenario.

Table 4.8: Comparison of modal choices for simulations A2, B2 and C2

	A2	**B2**	**C2**
Car	85.77%	51.13% (⇓ 34.65%)	53.05% (⇓ 32.72%)
Bike	14.13%	48.78% (⇑ 34.65%)	46.85% (⇑ 32.72%)
Walking	0.09%	0.09%	0.09%

4.3.2 Mobility of Individuals during the Activity of Leisure Trips

To demonstrate how agents can be reused across different domains, a second scenario has been created in which agents perform the activity of travelling to a music concert. New mobility concepts such as shared mobility services have led to more flexibility and new options in personal mobility and are intended to improve the use of available resources. However, the actual effects of these services are yet to be observed. In this research, simulation is given for an example scenario that looks at the impact of shared mobility on leisure trips. A specific type of shared mobility is *ridesharing*. Ridesharing is a term for organised carpooling, traditionally arranged among friends and family, but now commercialised as a service to connect individuals that have never met. Such services can help to reduce the number of private vehicles in use by encouraging individuals to share journeys occupying a single vehicle in place of two or more. The use of commercialised ridesharing depends on the mechanisms through which individuals are brought together. In this research, such mechanisms are referred to as the *pooling process*. There are different methods to implement the pooling process. One option is the use of auction mechanisms. Auctions are mostly known as a buying or selling process in which individuals place bids to purchase a particular item or service [249]. They can be used as an instrument to organise the access of individuals to the same limited resources. The use of such auction mechanisms in the context of mobility has been established [250]–[252]. Depending on how auctions are implemented (auction

Table 4.9: Indicators on environmental impact for simulations A2, B2 and C2

	A2	**B2**	**C2**
Global Travel Distance [km]	10828	10499 (⇓ 3.03%)	10117 (⇓ 6.57%)
Combustion Distance [km]	9790	6875 (⇓ 29.77%)	6368 (⇓ 34.96%)

Table 4.10: Normalised average traveller utility of changed agents for simulations A2, B2 and C2

	A2	**B2**	**C2**
Educational campaign	0.60634	0.5968 ($\Downarrow$ 1,57%)	-
Vehicle ban	0.60044	-	0.58316 ($\Downarrow$ 2.88%)

design), strategies of participating individuals may vary and therefore lead to different outcomes [253]. This suggests that a carefully implemented auction design can lead to an outcome favoured by the auction designer which in economics makes this a powerful instrument for guiding self-interested individuals towards social benefit and thus promote the use of shared mobility services. Finding an effective auction technique that optimises the sharing of journeys within a ridesharing scheme can be difficult. The use of a simulation tool allows the evaluation of auction mechanisms in silico before deployment into actual rideshare schemes.

Simulating this type of scenario requires the current simulation model from the grocery shopping example to be adjusted. Agents in the music concert scenario do not require knowledge about the food and supermarket domain which allows the activity ontology from the previous grocery shopping scenario to be removed. As agents in this scenario primarily deal with knowledge from the traffic domain, changes and extensions to domain knowledge only affect the travel ontology. Thus, the music concert scenario does not require a new activity ontology. The travel behaviour of individual agents has been extended to allow *ridesharing* to be included in their decisions. Relevant modifications for the music concert scenario primarily affected prejourney planning in which the agent makes a decision about its mode of transport. As an extension, ridesharing has been added as a new mode of transport $M \cup \{m_{ridesharing}\}$. With regard to utilities $u(\tau, m_{ridesharing})$, ridesharing may be treated as a private vehicle (e.g. car) given that the driver travels alone. The situation changes when an additional passenger joins the vehicle, to share the journey. Thus, values for $u(\tau, m_{ridesharing})$ are based on utilities of $u(\tau, m_{car})$ with deviations depending on the number of additional passengers. For each additional passenger, mode attributes for $m_{ridesharing}$ need to consider the following deviations as compared to the attributes of m_{car}:

- the utility on $flexibility$ decreases as changes to the journey have an immediate effect on the other passengers and therefore need to be taken into consideration.
- the utility on $time$ decreases as entry and exit of additional passengers as well as potential detours account for extra effort.
- the utility on $reliability$ decreases as there are more dependencies to be considered e.g. passengers being late or running into traffic jams due to additional detours.
- the utility on $privacy$ decreases as there are more passengers within the vehicle.
- the utility on $safety$ decreases due to unpredicted behaviour of passengers e.g. distractions.
- the utility on $environmental\_impact$ increases as emissions can be split among the driver and the passengers. However, if it wasn't for ridesharing, passengers might have chosen an even more environmentally friendly mode of transport, which is why the effect might be mitigated.
- the utility on $convenience$ decreases as there is less room for movement within the vehicle as well as storage space.
- the utility on $monetary\_costs$ needs to be handled specific to the scenario depending on who is travelling e.g. costs can be evenly split among friends, but is probably paid by the driver when they are driving members of their own family.

In addition to this, a new type of agent has been added to the simulation. Within the previous model that was created for the grocery shopping scenario, traveller agents select a mode of transport and then conduct their journey without the option to take additional passengers. The addition of ridesharing to the simulation model requires vehicles and travellers to be modelled as separate agents. Traveller behaviour needs to be extended to model passive passengers while vehicles must be able to contain information about passengers as well as the designated driver. In particular, traffic participants (travellers) hold relevant information about their journey (e.g. origin and destination) as the purpose of travel emerges from the individual. Furthermore, individuals can only use vehicles that are actually at their disposal, for example, vehicles that they privately own. Thus, traffic participants need to register vehicle agents to which they have access. Vehicle agents may vary in their passenger capacity depending on their type (e.g. car, truck, van, motorcycle). Based on this, vehicle agents need to record detailed information about which agents are inside the vehicle at any given time during the simulation. Figure 4.5 gives an overview of the information contained in the different types of agents.

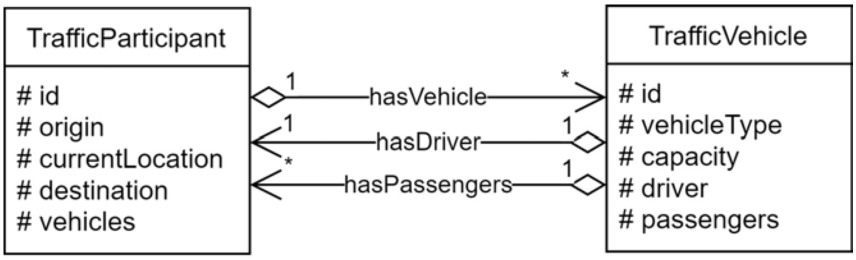

Figure 4.5: Separation of traffic participants and vehicle agents.

Ridesharing is typically organised either in private settings (among friends or family members) or through the use of commercialised ridesharing services. The former requires the implementation of social relations within the agent population. The *Barabási-Albert* algorithm can be used to model social structures and communities (see [254], [255]). The algorithm starts with a user-defined number of agents (σ_0) and iteratively adds new agents, thus creating a social network with eventually n agents. A new agent is connected to σ existing agents, where σ is a user-specified parameter, with a probability proportional to the number of connections within the existing agent population. As a result, agents with more connections have a higher probability of gaining new relations which leads to a social network in which there is a small number of agents with a high number of connections (hub nodes) and the majority of agents with only a small number of connections (satellite nodes). This process is also referred to as *preferential attachment*. However, in the real world new relations among individuals are often established in their immediate surroundings which is typically correlated to the geographic distance of their home location [256]. Consequently, agents in the simulation that are located in the same region should have a higher probability of knowing each other than agents that live farther away. Applying the standard algorithm to the agent population generates a social network that does not reflect this appropriately (see Figure 4.6). Thus, the Barabási-Albert algorithm has been modified into a two-step procedure. In the first step, the algorithm is applied to subsets of the agent population based on their home location and clustered by geographic regions. This produces a social network for each of these regions. In the second step, the algorithm is applied to establish transregional relations on the full set of agents. This time the user-specified parameter σ will be chosen to produce fewer connections as the probability for transregional relations should be smaller in comparison to the process of generating connections within the immediate surroundings (see Figure 4.7).

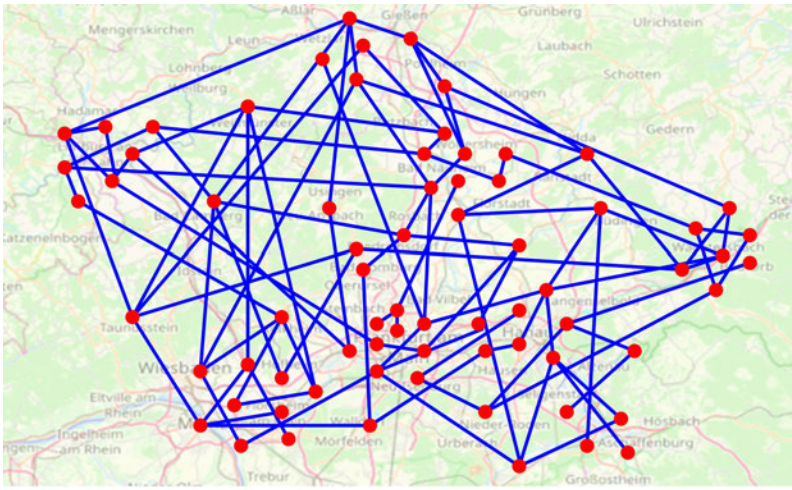

Figure 4.6: Example of a social network generated with the standard Barabási-Albert algorithm

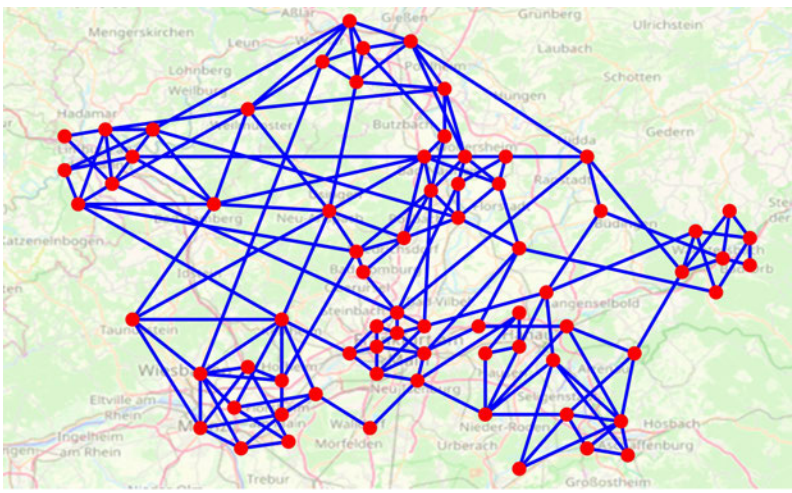

Figure 4.7: Example of a social network generated with the modified Barabási-Albert algorithm

Information on social relations is stored within the agents. Agents can use this information to arrange ridesharing in a private setting. Let A be the set of agents in the social network with $\Delta, P \subseteq A$. Δ is the set of drivers that contains agents that have chosen to travel with an individual vehicle (car) and P is the set of potential passengers containing agents that are looking for ridesharing options. Based on this, it can be defined $\Delta \cap P = \emptyset$. Furthermore, each $a \in A$ has a list of social contacts Λ_a. When dealing with ridesharing in a private setting, in this research it is assumed that agents connected through a social relation are in frequent contact and therefore are informed about the timetables and mobility needs of their friends and family members. Based on this, agents $a_1 \in P$ look in their list of social contacts Λ_{a_1} for potential drivers $\Delta_{a_1} = \Lambda_{a_1} \cap \Delta$, and successively request $a_2 \in \Delta_{a_1}$ sorted by shortest Euclidean distance $d(a_1, a_2)$ for whether a_2 would be willing to make a detour and give a_1 a lift. In the event that a_2 still has empty seats in its vehicle, the number of seats already assigned is used to determine the utility U_{TT} of both a_2 and a_1 to take a_1 as an additional passenger. Ridesharing is agreed when U_{TT} determines this to be the best option for both of them. Otherwise, the process continues for a_1 and alternative options are explored (finding another driver or changing to a different mode).

In contrast to this, commercialised ridesharing eventually causes interactions between unrelated individuals i.e. strangers. Agents have therefore been extended with an additional attribute that models their attitude towards travelling with strangers based on survey data provided by [242]. Interaction between these individuals is typically conducted through a digital service platform and thus is managed by the given processes of the platform. Connecting drivers and interested individuals is an essential task of these service platforms which can be implemented using auctions. To simulate the effects of different auction designs, the simulation has been extended with a central interface to flexibly plug in implemented auction algorithms. This interface requires a list of agents participating in the auction and returns the result of the auction i.e. a list of drivers with their assigned passengers. During the auction, agents submit bids according to the implemented mechanism of the auction to request a ride. Before submitting a bid, the agent verifies whether the utility U_{TT} for ridesharing still exceeds all of the alternative mode options based on the $monetary\_costs$ of the bid. In particular, let β be the amount of the bid to be submitted. β determines $monetary\_costs$ for the mode $m_{ridesharing}$ and thus has an effect on the mode attribute $u(monetary\_costs, m_{ridesharing})$. However, the computation of $u(monetary\_costs, m_{ridesharing})$ from β differs for every agent as β must be set in relation to the location where the agent wants to be picked up as well as the distance the agent wishes to be transported. For this purpose, β is compared to the costs of local taxi services $c(m_{taxi})$. Local taxi services typically charge a fixum based on the area of the pick-up location and the destination, as well as an additional fee depending on the actual travel time and driven distance. Estimated $c(m_{taxi})$ is used as a reference value for which the utility U_{bid} approximates 0 as it would be possible from this point on to simply call a taxi and forget about ridesharing. In addition to this, U_{bid} takes the maximised value on the utility-scale $u_{max} = 10$ if a ride turns out to be free of charge. Based on this, the following function was used to model the rapidly decreasing cost-benefit perception of individuals for U_{bid} as comparable to the approach used for modelling diminishing marginal utility (see Figure 4.8) (see [257]).

$$U_{bid}(\beta, m_{taxi}, u_{max}) = u_{max} * e^{(\beta * \frac{\ln(0.003)}{c(m_{taxi})})} \tag{4.5}$$

Furthermore, it is defined:

$$u(monetary\_cost, m_{ridesharing}) = U_{bid} \tag{4.6}$$

Computed utility $u(monetary\_cost, m_{ridesharing})$ is then used to determine U_{TT} for ridesharing. In the event that U_{TT} for ridesharing is expected to fall below the utility of an alternative mode option, the agent exits the auction and thus opts for a different mode of transport. Otherwise, the agent continues in the auction and submits its bid. The final costs of the ride are determined when the auction is completed.

As an example, simulation is given for a scenario situated in the German city of Gießen. Gießen is located within the Rhine-Main region which is part of one of the largest projects for on-demand mobility in Europe [258]. The project involves the launch

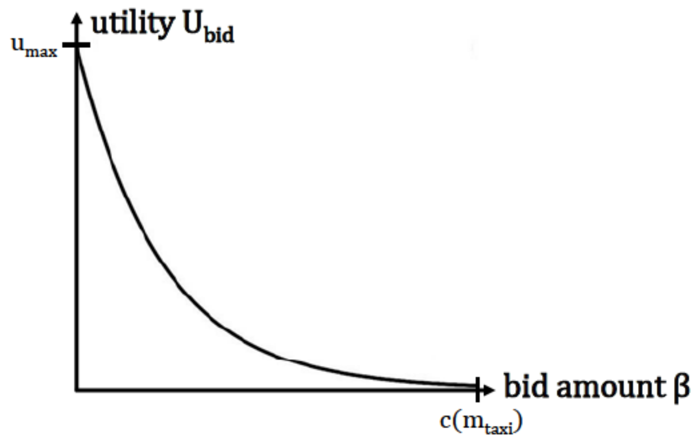

Figure 4.8: Function to compute utility of a bid

of a new commercialised ridesharing service. Based on this, different implementations of ridesharing are simulated for individuals travelling to a music concert. The music concert takes place at the local event venue *Hessenhallen* which is designed to accommodate 1500 visitors. Assuming that all the tickets for the concert are sold out, an agent population has been generated that replicates the empirical distribution of visitors. In particular, census information for persona profiles of the agents is based on survey data provided by [259]. Furthermore, a *Poisson* distribution has been used to model the preferred arrival time before the start of the concert. Visitors of the event typically travel from the surrounding area which is why a map has been chosen that covers the main residential areas (min. lat: 50.5291; min. lon: 8.5875; max. lat: 50.6257; max. lon: 8.7726). As visitors may come from every direction, agents have been randomly scattered over the map and it is assumed that agents start their journey from their assigned home location (see Figure 4.9).

Before the start of the journey, agents determine the mode of transport to travel to the event venue. Visitors who have a relatively short travel distance may be able to walk or cycle to the venue, whereas other visitors will have to rely on public transport or travel by car. This decision depends on the range of transportation modes that is available to the agent which was modelled using data from [242]. Furthermore, mode selection is based on the highest utility U_{TT} which is computed based on personal preferences a_τ as well as attributes of the mode $u(\tau, m)$ (see Equation 4.1). Analogous to the grocery shopping example, mode attributes in this scenario take utility values from a scale of

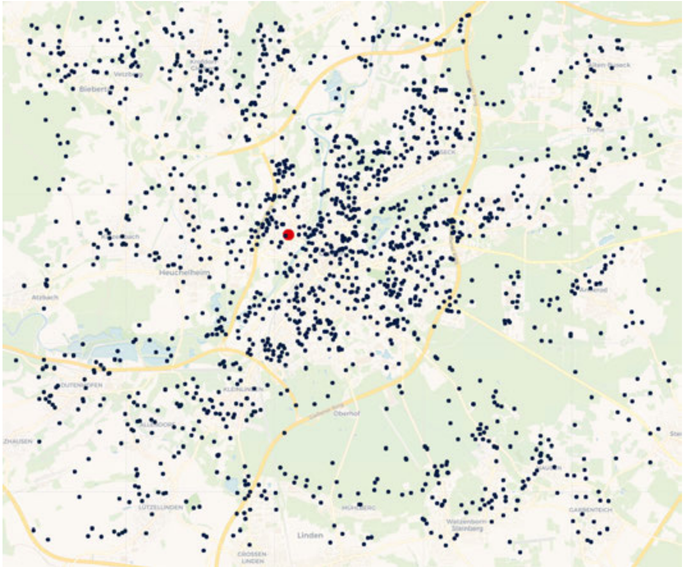

Figure 4.9: Agents scattered over the simulated area around Gießen

1 to 8, with 1 being the lowest and 8 being the highest utility. Utilities are based on values from the grocery shopping scenario (see Table 4.7) with some deviations that are due to the differing infrastructure of Wetzlar and Gießen. Gießen has followed a strategy in traffic planning to improve the traffic infrastructure for pedestrians and cyclists, which has led to the discontinuation of some road lanes and parking spaces for cars [260]–[262]. This changes the utilities of these modes in terms of reliability, safety and convenience (see Table 4.11). In Germany, public transport is typically included in the ticket for the event which in this scenario leads to $u(monetary\_costs, m_{publictransport})$ being maximised. Note that the focus of this experimentation is not to present a validated simulation model but to demonstrate how the proposed method allows agents to be reused across varying scenarios.

Individuals that travel by car may offer their friends a lift (private ridesharing). For this purpose, social relations among individuals have been implemented using the modified Barabási-Albert algorithm that uses a two-step procedure to generate (1.) social connections within the region and (2.) transregional relations. *k-means clustering* has been applied based on the euclidean distance of their home locations to obtain subsets of the agent population $\Gamma \subseteq A$ according to the 22 residential areas. For each of these subsets Γ a social network has been generated using the Barabási-Albert algorithm with $\sigma_0 = 5$, $\sigma = 5$ and $n = |\Gamma|$. In the second step, transregional relations have been

Table 4.11: Mode utilities $u(\tau, m)$ for the music concert scenario with $n \leq 4$ representing the number of non-driver seats in a car.

	Bike	**Walking**	**Public Transport**	**Car**	**Ridesharing for n add'l. passengers**
Flexibility	6	4	5	8	8 - n
Time	4	1	6	8	8 - n
Reliability	7	8	6	6 ($\Downarrow$ 1)	6
Privacy	5	4	1	8	8 - n
Safety	6 ($\Uparrow$ 4)	7 ($\Uparrow$ 1)	8	7	7 - n
Environmental Impact	8	8	6	0	0 + n
Monetary Cost	6	8	8	2	tbd from auction
Convenience	2 ($\Uparrow$ 1)	3 ($\Uparrow$ 2)	5	7 ($\Downarrow$ 1)	7 - n

generated within the whole agent population A with $\sigma_0 = 2$, $\sigma = 2$ and $n = |A|$. To reduce the number of individual vehicles, the event organiser encourages visitors that travel by car to not only limit ridesharing to their private surroundings but to also consider giving other visitors a ride in exchange for compensation (commercialised ridesharing). For this purpose, the organiser of the event provides a digital platform that connects drivers and individuals looking for ridesharing options via an auction system. Drivers can indicate their willingness to take additional passengers as well as the number of remaining seats. Interested individuals can submit a monetary bid to request a ride from one of these drivers.

As an artificial use case, this research looks at *whether different implementations of the pooling process can increase the use of ridesharing* and thus improve the load of passengers in vehicles. This would help to relieve the limited parking space at the venue as well as reduce the environmental impact caused by the event. For this purpose, three simulation runs ($S0$,$S1$,$S2$) have been performed with identical agent population. Note that the current implementation uses stochastic elements only while computing preferences a_τ, thus keeping the subsequent decision processes deterministic. This simplifies analysis of the use case, making comparison of simulations easier. $S0$ serves as a reference simulation and therefore features only privately organised rideharing and no commercialised ridesharing. In simulations $S1$ and $S2$, drivers first organise ridesharing in their private surroundings and in the case that a driver is willing to take additional passengers, the agent will participate in the auction process. $S1$ uses an *English auction* for the pooling process while $S2$ implements a *first-price sealed-bid auction*. In the *English auction* agents successively submit bids which raises the price until only one agent remains. Agents are allowed to bid multiple times until the highest bid wins. In comparison to this, the *first-price sealed-bid auction* allows agents to only bid once. Bids are submitted independently without any knowledge about their competitors. Same as in the *English auction*, the highest bid wins.

To measure the effects of the different implementations of ridesharing in this scenario, the following performance indicators have been altered or defined. The first indicator looks at the *avg. passenger load in vehicles* which is computed using arithmetic means over the number of agents travelling together in one vehicle. This indicator only considers the two transportation modes m_{car} and $m_{ridesharing}$. Furthermore, there are new indicators that measure the *number of privately organised ridesharing* as well as the *number of commercially organised ridesharing*. Analogous to the grocery shopping example, environmental impact is measured using performance indicators on aggregated travelled distances. In particular, *global travel distance* is computed as the sum of the overall distances travelled by the set of all agents. This indicator adds up the travel distance of each agent regardless of whether they were travelling within the same vehicle. In contrast to this, *combustion distance* only considers the two transportation modes m_{car} and $m_{ridesharing}$ as they produce additional exhaust gases. In this case, agents that travel in the same vehicle do not cause additional *combustion distance*. Public transport has been excluded from the calculation of this indicator, as rail and bus services generally operate regardless of the amount of passengers associated with the event.

As this is an artificial use case it can only be speculated about the results of the simulation. It should be noted that conclusions on behavioural changes require a well-designed research effort with field experiments which is not within the scope of this research. However, to demonstrate how the proposed model can be used to experiment on this type of scenario, a discussion is given of the simulation output for the artificial use case:

The comparison of modal choices in simulations $S1$ and $S2$ shows that the total amount of ridesharing increases in $S2$ (see Table 4.12). Drivers and passengers that participate in ridesharing can be examined separately. While the amount of ridesharing drivers show a slight increase, a more significant increase can be observed in the number of ridesharing passengers. Performance indicators in Table 4.13 reveal that this increase exclusively applies to *the number of commercially organised rideshares* as *the number of privately organised rideshares* remains the same. Thus, it can be concluded that the increasing use of ridesharing is the result of changes in the auction design. As the English auction in $S1$ allows agents to look into the bids of the others, agents only need to submit bids that are slightly higher than the others. This may lead to the final bid turning out to be lower than the winner would have been willing to pay. In contrast, as agents in $S2$ are limited through the *first-price sealed-bid auction* to only bid once without any knowledge of their competitors, agents are more likely to bid what they are actually willing to pay. As a result, bids in $S2$ tend to be higher than in $S1$ which increases the utility for drivers to accept additional passengers and thus leads to more rideshares. It can be noted that the indicator *average passenger load in vehicles* reflects this appropriately (see Table 4.13). Furthermore, it can be observed that the two indicators on *global travel distance* and *combustion distance* in $S1$ and $S2$ have increased in comparison to $S0$. This shows that promoting the use of ridesharing in this artificial use case does not necessarily improve environmental impact. One reason for this is that picking up passengers requires a detour which causes additional travel

Table 4.12: Comparison of modal choices for simulations $S0$, $S1$ and $S2$

Modal Choice	**S0**	**S1**	**S2**
Shared Ride (Driver)	14.53%	15.73%	18.40%
Shared Ride (Passenger)	14.53%	16.47%	24.07%
Walking	01.27%	01.27%	01.13%
Bike	05.67%	05.40%	04.53%
Public Transport	22.67%	21.93%	19.60%
Car	41.33%	39.20%	32.27%

Table 4.13: Comparison of KPIs for simulations $S0$, $S1$ and $S2$

KPI	S0	S1	S2
Average passenger load in vehicles	1.26	1.30	1.48
Number of privately organised rideshares	218	218	218
Number of commercially organised rideshares	0	29	143
Global travel distance [km]	8783.38	9170.63	9003.04
Combustion distance [km]	7229.51	7592.02	7570.00
Normalised average traveller utility	0.6201	0.6192	0.6091

distances. Another reason can be seen in the shift in modal choices when looking at which agents have actually switched to ridesharing (see Table 4.12). In particular, the number of pedestrians, cyclists and individuals that use public transport has decreased. Furthermore, the number of car drivers also decreases as they are counted as *ridesharing drivers* when taking additional passengers or switching to being *ridesharing passengers*. All in all, results show that rather than getting visitors to abandon their private vehicles, ridesharing in this scenario has served as an alternative to more environmentally friendly options which has led to counterproductive effects. Another observation is that as the amount of ridesharing increases, the average traveller utility decreases. In this scenario, this is caused by agents submitting a bid for ridesharing based on the number of passengers known at the time of the auction. Thus, agents assume a utility value for ridesharing that may decrease if at a later point in time more agents want to join in the same vehicle. However, once a ridesharing agreement has been made, the ride is considered confirmed and typically will be completed as it would be in the real world. This can lead to agent utilities being lower than initially expected.

This concludes demonstration of reusing agents for the music concert scenario. The purpose of these experiments is to demonstrate how the proposed framework is able to capture plausible changes in performance indicators when using different input settings. Validation for exact values of simulation results typically involves empirical validation against real-world data which currently is not the focus of this work. For example, results of this experimentation may have differed based on the implementation of the social network (see [263]). For real-world applications, relevant data for input and validation of the simulation includes geographic information of the simulated area as well as census data and behavioural surveys. With regards to geographic map information, this data can be obtained from OpenStreetMap [79]. In addition to this, census data is typically accessible through governmental institutions whereas obtaining survey information on activity-specific behaviour can be more difficult. In particular, this information is usually surveyed as part of consumer studies by either public or private research institutions. Primary data and results from these studies are occasionally published in online databases such as GESIS[1], Statista[2], or can usually be made available upon request or individual agreements.

4.4 Summary of Chapter 4

This chapter looked at the issue of finding appropriate methods to model individuals and their decision behaviour for current scenarios in mobility. For this purpose, section 4.1 provides a brief introduction to semantic technology. In section 4.2, a framework is presented that uses semantic technology to capture knowledge and preferences of individuals as determining factors of agent decisions. The application of semantic technology allows the implementation of general agent activity logic to be separated from aspects of modelling agent knowledge. Based on the three layers of the CommonKADS approach, the traditional BDI agent has been extended with a qualitative model of world knowledge. The lowest layer contains information on domain knowledge and abstracts common concepts from the travel domain from activity knowledge. Based on this, activity knowledge can be flexibly extended or replaced which allows agents to be reused across different scenarios. In the second layer, this domain knowledge is extended by person-related concepts that describe the agent attributes. In particular, census properties from this ontology serve as input to SWRL rules that compute agent preferences based on survey data. Information from the first and second layers is used for agent decision-making which is implemented using BDI agents in the third layer.

For demonstration purposes, section 4.3 demonstrates simulation for two mobility

[1] `https://www.gesis.org/en/home` (access on 26/04/2023)
[2] `https://www.statista.com/` (access on 26/04/2023)

scenarios that differ in the considered travel activity. The first scenario deals with traffic caused by individuals during their grocery shopping while the second scenario simulates individuals travelling to a music concert. As stated previously, the focus of this research lies not in the input values given in Tables 4.3 and 4.7, but in the manner in which the simulation responds to their modification. In the experiments for the grocery shopping scenario, modifications model an improvement in the quality of alternative travel modes to car travel, allowing walking and cycling to be viewed more favourably as genuine alternatives to the car. It can be noted that the simulation output reflects this appropriately. Table 4.8 demonstrates a shift away from car travel when the utility of the other modes is increased and policies that favour them are introduced (B2 and C2). It is this ability to respond appropriately to differing input scenarios that makes the proposed methods in this research valuable. Effects on individuals are the basic cause of how the system changes under interventions which is why they cannot be ignored. Conducted experiments have shown how modelling more details of individual behaviour establishes the basis for measuring individual utility and thus creates an indicator for measuring the effects of policies not only on global system behaviour but also on individuals. To demonstrate how agents can be used across varying scenarios simulation has been given for a second scenario that looks at different implementations of ridesharing in traffic on leisure trips.

Chapter 5

Current State in Validation and Verification of Agent-based Simulations

Given that multi-agent simulations model systemic behaviour based on a large set of individuals, validation and verification of these simulations can become particularly complex. Thus, validation and verification may cover a wide spectrum of methods that focus on different aspects of the simulation. This chapter gives an overview of the current state of validation and verification specifically of agent-based simulations. The focus of this chapter is to reflect on important related work and to outline current challenges and limitations. Based on this, specific issues can be addressed in the scope of this research to generate more insights into the internal mechanisms of the simulation and thus make a contribution towards improving the validation and verification of agent simulations. For this purpose, it is important to clarify the different approaches to the validation of multi-agent simulations. [29] have described different levels of validity/validation for multi-agent simulations which are consistent with discussions given by [264] and [265]:

1. ***Replicate or empirical validation*** is the most basic concept which looks at whether inputs and outputs of the simulation match observations of the real-world system. In this case, simulation output is compared against historical data typically over multiple runs.

2. ***Predictive validation*** goes one step further and looks at whether the simulation

is able to forecast future behaviour. In this case, simulation output is compared to field experiments on the real-world system.

3. ***Structural validation*** examines how results are computed during the simulation. Simulation output may correctly reflect patterns and findings from empirical data but might have been obtained through a completely different process.

While *replicate* and *predictive validation* primarily examine the correctness of inputs and outputs of the simulation which resembles *black-box testing* from software engineering, *structural validation* looks into the internal mechanisms of whether results are computed through a plausible process similar to *white-box testing*. Furthermore, replicate and predictive validation can be considered as part of *operational validation* whereas structural validation is covered by *computerised model verification*. Validation techniques from operational validation, such as sensitivity analysis or using visualisations and statistical methods for crosschecking simulation output against observational data, are also used for replicate and predictive validation of agent-based simulations. Assuming that appropriate validation data can be obtained, the course of action for replicate and predictive validation is well understood. However, the situation changes when dealing with the structural validation of agent-based simulations. This type of validation is known to be particularly difficult and remains an ongoing challenge. This research therefore looks into the challenges of structural validation as part of computerised model verification to be further researched.

Formal verification and *testing* are both activities from computerised model verification that share the objective of finding errors (bugs) in the implementation of a software program [266]. The difference between these two activities is that *testing* can only find errors, but not prove their absence, while the idea of *formal verification* is to provide mathematical proof of correctness and thus prove the absence of errors [266]. However, providing proof of correctness is complex and in some cases not even possible. For example, the *Turing halting problem* (see [267]) is undecidable producing an infinite set of states and therefore it is not possible to provide formal verification [268]. For this reason, practical application in computerised model verification tends to follow the principle of falsification (see [269]) rather than providing formal proof of correctness. Thus, practical computerised model verification is to be understood as a form of *testing* rather than *formal verification* in the mathematical meaning.

5.1 Model Checking

Model checking is a branch of *formal methods* that uses mathematical techniques to verify the correctness of a software program. This involves using mathematical models to describe the behaviour of the system which can then be used to apply logic-based reasoning algorithms in order to prove that the system satisfies a set of specified properties. In essence, model checking can be broken down into the following activities [268]:

1. ***Modelling*** a finite state-transition graph (also known as the Kripke structure) to obtain a formal representation of the software program. This assumes that the program can be represented as a graph with finite states and transitions in the first place.

2. ***Formal specification*** of properties that define what is considered correct program behaviour. Using modal logic these properties are typically expressed as theorems that should not be violated at any time during the execution of the program.

3. ***Applying algorithms*** to explore the state space of the graph to identify whether there exists a state in which the software program violates a property defined in the formal specification. The program is assumed to be correct when the algorithm has processed all states and no violation is found.

Model checking is often referred to in the literature as a means for formal verification. However, in practical application verification guarantees provided by model checking are limited as it is performed only for a specific set of properties while typically requiring assumptions about the environmental conditions [266]. For example, the Turing halting problem is undecidable and therefore cannot be modelled as a finite state-transition graph. To do so, assumptions are required about the environmental conditions of the program. In particular, integer values need to be bounded in order to constrain potential states to a finite space [268]. These assumptions about the environment are often necessary to make a program verifiable with model checking. However, this leads to verification guarantees to apply only to a subset of the potential state space. Thus, rather than being a means for formal verification in the strict mathematical meaning, model checking should be viewed as a systematic and more exhaustive variation of testing as it is able to find bugs within a finite state space but cannot prove the absence of bugs when potential states are infinite [266].

The approximation and abstraction of infinite state systems is one of the main challenges in research on model checking [268]. In this context, a closely related issue is the state-explosion problem [270]. In simple programs that for example involve multiple interacting parties, it can be demonstrated that the potential state space increases exponentially with the number of parties and actions (strategies) involved. Processing these large state spaces, particularly with model checking and reasoning algorithms requires a lot of computing capacity which is expensive and in some cases makes it infeasible to be applied in practice. Research in model checking has since been trying to cope with the state-explosion problem from different perspectives. For example, improving algorithms to search/traverse the state graph (*directed model checking*) [271], [272], reducing complexity of the state graph by removing redundancies (*partial-order reduction*) [273]–[275] or dividing the problem into smaller subentities to deal with (*compositional model checking*) [276], [277].

Further research on model checking deals with the formal specification of properties in multi-agent systems based on modal logic [278], [279]. [280] have given a detailed discussion on the use of modal logic for the formal specification of multi-agent systems.

The application of model checking techniques to find errors and bugs in software systems comes with several benefits. Model checking algorithms are applied to the finite state-transition graph of the implementation (Kripke structure) which is an abstract representation of the source code. As a consequence, it is not required to have the actual implementation of the model, given that such a graph structure could already be obtained based on the conceptual model. Thus, model checking can be employed at a very early stage in the model development process during conceptual model validation which allows errors and bugs to be avoided before they are implemented. Another benefit is that the approach to model checking is formalised and mathematically founded. It is therefore possible to fully automate the verification process e.g. [281]. Furthermore, model checking is able to prove the absence of errors by systematically exploring the finite state space. This makes it possible to locate errors and bugs even for edge cases which otherwise would have been difficult to find using conventional testing methods. However, this thorough approach of model checking also leads to significant drawbacks in performance. The main challenge continues to be the scaling of model checking from toy examples to real-world problems. In particular, for problems that involve a large number of agents with a wide range of actions, the combinatorial complexity leads to state space explosion. In such cases, model checking becomes time-consuming and requires a lot of computing capacity. This is particularly noticeable when dealing with agent-based traffic simulations. The application of model checking for real-world problems can therefore be impractical which is why model checking is mostly employed for safety-critical systems. Furthermore, simulated traffic scenarios typically feature a level of complexity that makes obtaining the Kripke structure as well as a formal specification of properties difficult and in most cases incomplete. This lack of formalism when dealing with agent-based traffic simulations makes model checking ineffective while requiring substantial resources. As a consequence, model checking in practice is not particularly suited for computerised verification of agent-based traffic simulations on real-world scenarios. In such cases where model checking cannot be applied effectively, verification of software systems is achieved through appropriate testing procedures.

5.2 Static and Dynamic Testing

Testing is an important activity mostly known from software engineering. It has been described as the process of executing a program with the intent of finding errors which should be conducted throughout all stages of software development [282]. Testing is a less formal approach to computerised model verification than model checking as it does not systematically explore the state space and thus cannot prove the absence of errors [266]. In return, testing can be considered a more lightweight method which can be faster and less resource intensive. The quality of testing ultimately depends on how test cases are determined. As mentioned in chapter 2.3.3, there are two types of testing related to computerised model verification: *static* and *dynamic testing*. [70] has given an overview of commonly used techniques for both *static* as well as *dynamic testing* of simulations. The difference between these techniques is that *static testing* examines the system design without needing to execute the simulation whereas *dynamic testing* checks the computerised model from a behavioural perspective when it is being executed [69]. For example, common approaches in dynamic testing include animation, tracing or sensitivity analysis to investigate the behaviour of the simulation. Animating the movements of simulated entities can help to identify behavioural patterns and facilitate the interpretation of simulation output. Tracing is a technique in which the behaviour of specific entities is followed throughout the simulation to determine whether the logic of the model is correct [69]. Sensitivity analysis investigates the output behaviour of the model by executing simulations under different input conditions and looking at how outputs change.

In contrast to this, static testing looks at the system design for example by manually reviewing the source code or dealing with graphical representations of the system (e.g. UML diagrams [13] or cause-effect graphs) [9]. Software engineers can perform testing activities only to a certain extent on their own. To improve the testing of simulations, experts from the simulated domain are typically consulted. Getting domain experts in to verify the plausibility and correctness of logical coherences in the system is also known as *face validation*. However, these domain experts do not necessarily have a background in computing which is why face validation requires software engineers to walk the expert through the source code (structured walkthrough). This approach is based on manual processing and thus can be time-consuming. Furthermore, miscommunication between domain experts and software developers is a regular occurrence which may lead to errors, and important details getting lost in the process [283]. Face validation that is complemented with additional graphical models of the system can help to overcome these issues. Graphical models are a formalised and compact representation of the system. There is a broad spectrum of graphical approaches to model software systems that differ in the scope of their modelled subject e.g. UML class vs activity diagrams. [284]–[288] have extended UML diagrams with agent-related concepts some of which have been implemented as tools to guide the software developing process of multi-agent systems [289], [290] and even dealt with system design from a more behavioural perspective [291], [292]. In this context, another type of graphical model that also addresses the behavioural perspective on system design is event graphs [293], [294]. While these approaches describe the software system from different perspectives, cause-effect graphs are particularly relevant when it comes to modelling relations between input and output variables [295]. Relations between input and output variables are of special interest for policy-making to determine effective interventions. Furthermore, graphically representing these relations can help provide more insight into how output variables are computed. Thus, the use of cause-effect graphs can provide additional information for structural validation of simulations.

Cause-effect graphs have been applied in non-agent-based simulations, where they model relations between variables on a common level of abstraction [10], [293]. For example, figure 5.1 demonstrates a causal loop diagram, which is a specific variation of cause-effect graphs, that models the mutual influence between the two variables: the amount of *predators* and *prey* in an ecological system [10]. Increasing the numbers of predators (foxes) leads to a decreasing population of prey (snow hares), while at the same time, decreasing numbers of prey leads to food scarcity for predators, which reduces their population. This allows the population of prey to recover, resulting in a

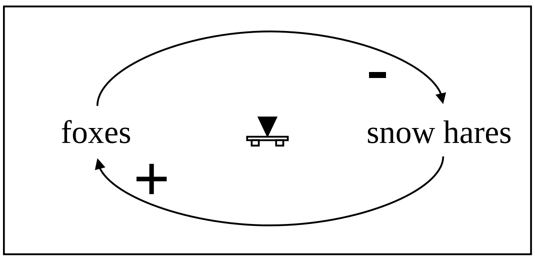

Figure 5.1: Example of a causal loop diagram on the mutual influence of predators and prey [10].

causal loop given that the system stabilises itself within certain limits.

However, due to the missing link between the micro- and macro-system level of agent-based simulations (also known as the multi-level property of agent-based models [296]), conventional cause-effect graphs need to be extended to deal with the hierarchical structure of cause-effect relations in agent-based simulations. Developing a tool that is able to extract cause-effect relations from a given computerised simulation model, can help achieve a graphical representation of the simulation and therefore can be a significant contribution towards improving structural validation of agent-based simulations based on face validation.

5.3 Summary of Chapter 5

This chapter has given an overview of the current state of validation and verification of agent-based simulations. In particular, literature distinguishes between empirical, predictive and structural validation. Activities for the validation of agent-based simulations typically rely on approaches from empirical and predictive validation that primarily look at the correctness of inputs and outputs. However, an ongoing challenge remains in the structural validation of agent-based simulations where the focus is to determine whether results are obtained through a plausible process. Thus, structural validation focuses on computerised model verification rather than operational validation of simulations. Common techniques from computerised model verification involve formal verification and testing. While testing aims at finding specific cases in which a software program behaves incorrectly, formal verification seeks to prove the absence of errors. However, formal verification can be difficult as even simple programs can lead to an infinite number of states and therefore formal verification cannot always be achieved. Hence, practical applications focus on model checking rather than trying to achieve true formal verification.

In section 5.1, model checking is presented as a structured approach to testing trying to prove the absence of errors within a finite state space. For this purpose, model checking relies on assumptions to make the state space finite. Within this finite state space, the program can then be checked against a set of defined theorems (formal specification). However, state spaces of software programs easily become very complex and thus model checking typically requires a lot of computing capacity. For this reason, model checking can be limited when it comes to the computerised verification of agent-based simulations, especially when looking a large simulations for real-world scenarios.

In cases where model checking is not feasible, computerised verification is achieved through testing. In section 5.2, a differentiation is made between static and dynamic testing. Static testing examines the system design without needing to execute the simulation whereas dynamic testing checks the computerised model from a behavioural perspective when it is being executed. There are methods from both dynamic and static testing that can be used for the computerised verification of simulation models, e.g. tracing or face validation. However, these methods typically require a significant degree of manual processing e.g. software developers manually reading the source code together with domain experts. Structural validation using these methods is prone to errors and therefore requires further research.

Chapter 6

A Graph-based Framework for Extracting Cause-Effect Relations from Agent-based Traffic Simulations

As described in this thesis, agent-based simulations have been applied to study and predict social-behavioural patterns in traffic. They are particularly suitable as they allow system behaviour to be modelled and simulated as the emergent result of behavioural patterns of a large set of individual travellers. While this type of modelling can be more intuitive, the approach also leads to new challenges in the reproduction of results ex-post as well as comprehension of cause-effect relations, as outcomes are achieved through the collective of autonomous decisions. Therefore, users applying agent methods may be left in doubt about the quality and validity of the results and thus cannot argue convincingly using the predictions (see [297] for a discussion on trust in simulations). At the same time, transportation researchers that come up with a conceptual model for simulating specific traffic problems, work together with software engineers to build computerised simulations but may end up with deviations in the implementation due to peculiarities or subtleties of programming. It is currently difficult for these transportation researchers, who may have little or no background in software development, to understand and validate the internal structure of the implemented model (structural validation). This makes it difficult to trust the findings obtained from these simulations. [298] have discussed the role of explanations for building trust in intelligent systems with emerging behaviour. Trust in intelligent systems can be promoted by increasing the transparency about how results are derived. This can help to make the results of simulations more convincing and thus motivates the effort to extend agent-based simulations with more explanatory capabilities.

Based on this, this chapter focuses on creating more transparency about the internal mechanisms of agent-based traffic simulations. In particular, by providing additional information about how results are derived. Explanations can be given through different presentations, such as textual descriptions or graphical representations. Informational focus, level of detail as well as presentation may vary depending on the target audience for which explanations are being provided. For example, software developers may be interested in the interaction and information exchanged between system components (e.g. for debugging purposes) whereas policymakers need to understand cause-effect relations from a social-behavioural point of view to identify effective and efficient interventions in the traffic system. Effects of interventions (system behaviour) are typically captured through performance indicators (output variables) that measure relevant information to answer specific research questions while causes are determined by modelled user input (input variables) or resulting interim variables computed during the simulation. Therefore, this research focuses on retrieving more explanatory information about the cause-effect relations between input and output variables in agent-based simulations. For this purpose, this research proposes a graph-based framework that automatically extracts relevant information from the simulation and generates a graph-based representation of cause-effect relations from input variables on the individual level (e.g. preferences and attributes of traveller agents) to performance indicators of the system on the global level (e.g. overall transit times or environmental impact). Note that the intention of this thesis is to automatically extract and formally represent cause-effect relations in a graph structure for improving the explanatory capabilities of agent-based traffic simulations and not to present a comprehensive method for structural validation. However, the graphical representation of cause-effect relations can serve as additional information for domain experts to get an overview of the system and thus can be considered a contribution towards improving structural validation based on face validation. An earlier version of parts of this chapter has been published in [2].

6.1 Method

Cause-effect graphs have been applied in non-agent-based simulations, where they are used to model chained causal relations between input parameters and system behaviour measured by appropriate indicators (e.g. [10], [293]). However, multi-agent simulations shift the paradigm of chained causal relations to multiple levels of detail and abstraction. Consequently, there is a need for conventional cause-effect graphs to be extended. To capture the hierarchical structure of cause-effect relations between input and output variables in agent-based traffic simulations, this chapter proposes a new graphical modelling method that is called *Multi-Agent Modelling Notation (MAMN)*. Based on this, relevant information about cause-effect relations can be automatically retrieved from the simulation by implementing appropriate logging mechanisms using techniques from *aspect-oriented programming (AOP)* [299] and represented in graph structures. This allows more insights to be given about the internal mechanisms of the simulation.

6.1.1 Applying Cause-Effect Graphs to Agent-based Models

System behaviour, which is often mirrored in output variables (key performance indicators), depends on both the modelled input as well as the internal mechanisms and dynamics of agent behaviour. Crucially, direct and indirect consequential effects are the outcome of calculations according to mathematical functions and formulas expressed in the model. Chaining these calculations and corresponding intermediate variables reveals causal relations between input and output variables. These relations can be complicated in multi-agent systems, but representation in a graph structure can make the internal mechanism of the model more accessible.

Cause-effect graphs have previously been used for a number of purposes, including software testing [300], system dynamics models [301], and management tools [302]. They are an explicit and precise formalisation of logical systems and serve as a compact representation. For example, cause-effect graphs have been used in software testing to specify test cases for combinations of input and output variables [300]. Input variables define causes, while effects are represented as output variables. Cause-effect graphs have also been applied for visualising aspects of simulation models [10], [293]. These approaches typically differ in the type of cause-effect relations modelled in the graphical representation, e.g. [293] focuses on modelling cause-effect relations between abstract events, rather than the computational aspects of key performance indicators. However, cause-effect relations between input parameters and performance indicators are of particular interest for policy-making. As presented in Chapter 5.2, a specific variation of cause-effect graphs are causal loop diagrams which are used in system theory to model mutual effects between variable entities, e.g. mutual influence between predators and prey in an ecological system [10]. The system theoretical view of reducing the complexity of information from reality to formal systems is an essential prerequisite for building computable simulation models. Building richer simulation models typically involves the modelling of more system variables. Thus, observed effects are not a direct consequence of a single variable but of multiple causative variables (or chains of variables). *Bayesian networks* are an example of cause-effect graphs that allow output variables to be linked back to possible (chains of) input causes based on probabilities [303], [304]. However, applying cause-effect graphs to agent models has been difficult due to causal relations being the emergent result of behavioural patterns of a large set of individuals which changes the paradigm from chained causal relations to several levels of detail and abstraction. This implies that the use of graphs working on a single level of abstraction is not appropriate and that a hierarchical approach separating individual and global perspectives would be more suitable. The semantics of a graphical notation need to capture the internal aspects of individual agents, i.e. preferences and their decision-making behaviour, as well as their context in the computation of performance indicators at the global system level.

Other approaches to the visualisation of multi-agent systems have focused primarily on system design by extending traditional methods from software engineering (e.g. [284], [286]). Some of these methods have been implemented as tools to guide the software developing process of multi-agent systems [289], [290] and even dealt with system design from a more behavioural perspective [291], [292]. However, these modelling methods have a primary focus on the technical design of software components, rather than the cause-effect relations of performance indicators in a simulation model. Hence, there is a need for a graphical method that is able to capture exactly this type of causal relations between input parameters and performance indicators on the social-behavioural level as these are particularly relevant for policy-making.

For this purpose, this research proposes a new graphical notation to model the hierarchical structure of cause-effect relations between input and output variables in multi-agent simulations. The focus is on modelling the main logical constructs commonly used to simulate agents in route choice scenarios. As input variables in multi-agent simulations mainly revolve around individuals and their behaviour, this should be reflected in the modelling. Balke and Gilbert have given an overview of established architectures used in literature for modelling agent behaviour [89]. For this research, focus is given to modelling agent behaviour according to the commonly used *Belief-Desire-Intention (BDI)* model [101].

Unified Modeling Language (UML) is a standardised general-purpose modelling language to visualise the design of a system [13]. It includes various types of diagrams that describe a system from different perspectives (see Figure 6.1). For example, *composite structure diagrams* and *class diagrams* can be used to visualise the *structural* relations between system components while *state* and *activity diagrams* are used to model the dynamic *behaviour* of objects within the system. However, research and practical application typically lead to new and distinct use cases that cannot always be completely covered by UML. For this reason, UML allows the extension of its diagrams by creating dedicated *profiles.* A profile allows metaclasses of a UML metamodel to be adapted for different purposes [13]. This implies that new syntax constructs as well as new semantics can be added to a graphical model by creating instances of the

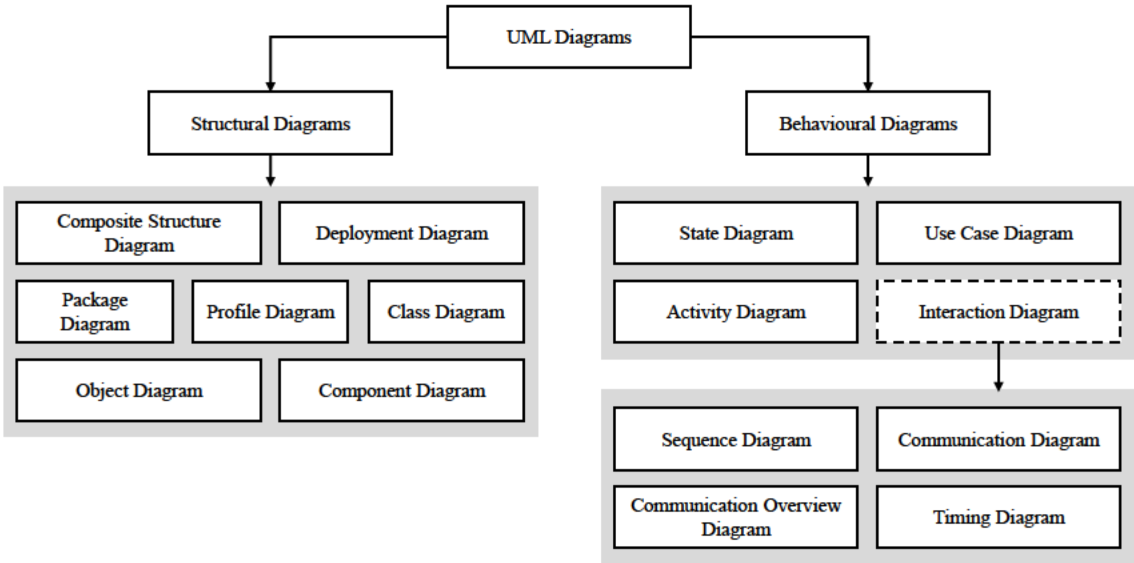

Figure 6.1: Types of UML Diagrams (see [11]).

given metaclasses. As the purpose of MAMN is to capture the cause-effect relations of input and output variables in agent-based simulations, the metamodel of a behavioural diagram such as the UML activity diagram can be particularly suitable (see Figure 6.2). In particular, the progression of computational functions, that ultimately process the user input to determine the resulting intermediate as well as output variables, can be modelled as activities (`actions`). Furthermore, UML activity diagrams allow the input and output of activities to be modelled as `parameters`. Thus, the essential prerequisites are given to model the cause-effect relations of input and output variables based on UML activity diagrams. As the context of variables in the simulation is important for understanding cause-effect relations, the standard profile of UML activity diagrams needs to be extended with a structural view. For example, agent preferences can have a significant impact on travel behaviour which consequently has an effect on performance indicators of the traffic system. In this case, the structural view is required to model the context of preference variables as part of the agents. Hence, modelling elements of MAMN have been defined based on the metamodel of UML activity diagrams and to comply with the UML notation standard. Figure 6.3 gives an overview of the main notation elements used in MAMN.

Formally, let $G = (V, E)$ be the directed labelled MAMN graph with vertices V and edges E. $V = F \cup N$ is a heterogeneous set of vertices with F the set of `functional` nodes

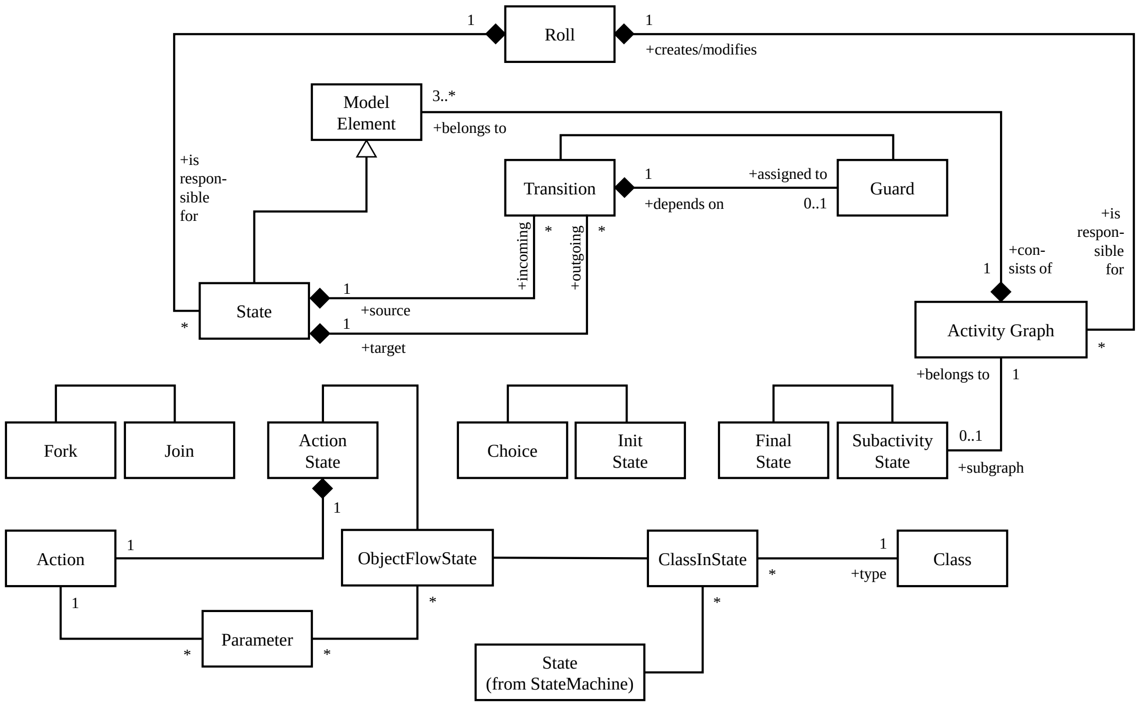

Figure 6.2: Metamodel of Activity Diagrams (see [12], [13]).

and N the set of `variable` nodes. Vertices $f \in F$ are equivalent to notation elements known from UML activity diagrams and are used to model aggregations of functional sequences as well as the start and end of simulation (sub)processes. As the focus of MAMN is on modelling cause-effect relations of input and output variables, functional nodes $f \in F$ serve as an abstraction of the implemented logic in the simulation program. This includes for example mathematical formulas or algorithms used to compute agent decision-making of which the result is reflected in the output variables. The abstraction of these logical sequences into functional nodes allows cause-effect relations between variables on the same level to be modelled as input/output chains while at the same time details of functions can be shifted to a sub-level as a separate graph. For this purpose, UML activity diagrams capture details of functions as subactivities. Functional sequences that are moved to a sub-level are indicated on the upper level by adding the subgraph symbol to the activity node. Vertices $n \in N$ are new instances of the `parameter` metaclass defined in the metamodel of UML activity diagrams. These vertices are an extension to the standard profile that adds a structural view of simulation variables. In particular, this type of vertex is used to model input parameters and performance indicators of the simulation, as well as relevant intermediary variables that are produced during the computation of performance indicators. The shape of variable nodes $n \in N$ depends on the contained type of information e.g. `primitive` (rectangles) or `complex` information (circles). The outline of these variable nodes indicates whether n is a `single` variable (solid) or a `collection` of variables (dotted).

Furthermore, MAMN includes dedicated nodes for modelling the internal aspects of agents based on the BDI model. BDI agents typically perform action decisions (intentions) on the basis of defined goals (desires) and their modelled knowledge of their external world (beliefs) [101]. Based on this, MAMN introduces *mental-level* nodes N_M and F_M. `Beliefs` are variables N_M that contain information about the current internal state of an agent as well as perceived information about its surrounding environment. This is modelled using a circle element with a bold dashed border. In addition to this, `intentions` and `desires` are functional nodes F_M that model agent behaviour. They are instances of the `action` metaclass defined in the metamodel of UML activity diagrams. `Desires` define the goals of an agent to maintain or achieve a certain state. In the context of mobility, this can also be referred to as *travel purpose*. Mobility of individuals typically is a necessary means for pursuing personal objectives, such as travelling to work or going to shop for groceries. In the MAMN notation, desires are modelled using a trapezium shape. To achieve a desired goal agents have to perform actions or a series of subsequent actions. This is referred to as `intentions` in the BDI model and modelled in MAMN as a hexagon. These new notation elements (trapezium and hexagon) have been chosen based on perceptual discriminability (see [305]) as well as with regard to the symbol being a basic geometric shape so that they can be implemented for the visualisation by common graph frameworks.

In addition to this, vertices of a graph are linked through edges $e \in E$ which are instances of the `transition` metaclass defined in the metamodel of UML activity diagrams. Edges are pairs (v_1, v_2) with $v_1, v_2 \in V$. The standard `flow transition` of UML activity diagrams is interpreted in MAMN as a causal relation. However, as vertices have been extended with a structural view, edges will also be extended with the appropriate semantics to model relevant structural relations between variables. Let there be $E = E_{Causal} \cup E_{CausalForEach} \cup E_{Constructor} \cup E_{Selector} \cup E_{Contain} \cup E_{ContainForEach}$.

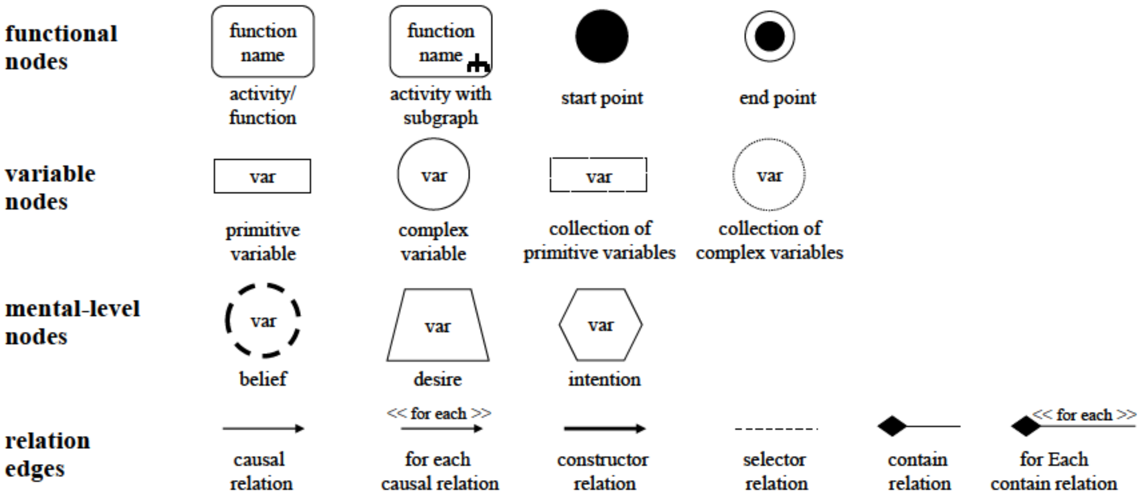

Figure 6.3: Revised MAMN Notation Elements.

$e \in E_{Causal}$ define causal relations for which applies v_2 is causally dependent on v_1. While the standard profile of UML activity diagrams typically uses this type of relation to link activity nodes, in MAMN this type of relation is also used to model input and output variables of activities. In the case of v_1 being a `collection` node $v_1 \in N$, activities that are executed for each of the elements in the collection can be initiated by an edge $e \in E_{CausalForEach}$. Apart from this, edges $e \in E_{Constructor}$ can be used to model a `constructor relation` in which v_2 is created from v_1. In this case, v_1 must be a functional node $v_1 \in F$ that results in a variable node $v_2 \in N$. Edges $e \in E_{Selector}$ model a relation in which one item is being `selected` from a collection which serves as input to a function. Thus, v_1 must be a `collection` node $v_1 \in N$ and v_2 a functional node $v_2 \in F$. Finally, the last type of edges $e \in E_{Contain}$ defines a relation in which a complex variable node $v_1 \in N$ `contains` the information of $v_2 \in N$. This relation is equivalent to the `composition` relation found in UML class diagrams. In the case that v_1 is a `collection` of complex variables, $e \in E_{ContainForEach}$ can be used to express that each item in v_1 holds its own information v_2. This concludes definitions for the MAMN graph structure.

6.1.2 Extracting Relevant Information to Generate Cause-Effect Graphs

Formalisation of cause-effect relations between input and output variables as MAMN graphs serves as a compact representation of the simulation model. Graph structures can be utilised in a bi-directional process to either transfer a theoretical simulation model into a concrete implementation as an executable piece of code (*forward engineering*) or to represent information from a given implementation (*backward/ reverse engineering*) which can be used to increase transparency and explainability of a system. In this section, the interest of this research lies in the second manner. In particular, MAMN graphs that are generated automatically from the simulation can provide more insight about how the results of agent-based traffic simulations are derived and thus can give explanatory information about the internal mechanisms of the simulation. For this purpose, the following three-step approach has been developed in the scope of this research (see Figure 6.4).

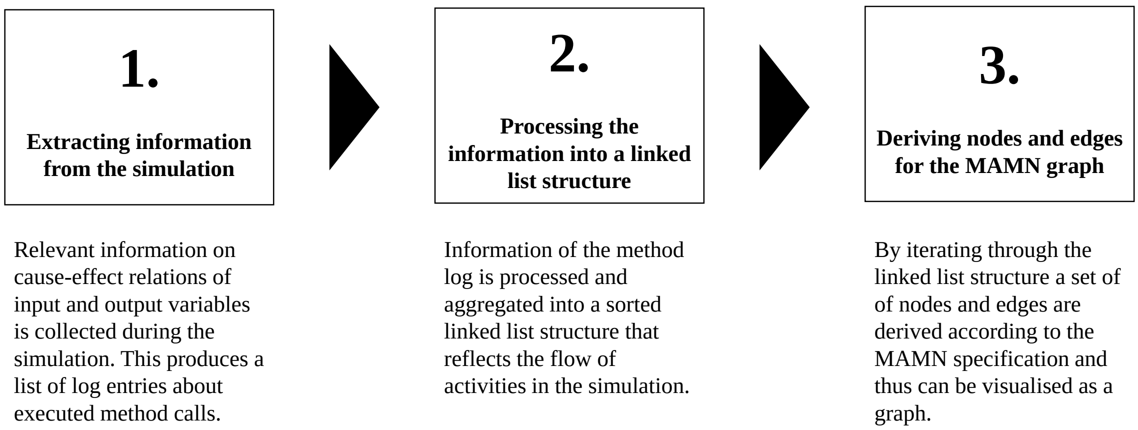

Figure 6.4: Three-step approach for generating MAMN graphs.

In the first step, relevant information for the MAMN graphs needs to be collected during the simulation. As MAMN graphs are extensions of UML activity diagrams, the progression of how variables are processed is reflected in the method calls executed during the simulation. Hence, information about method calls needs to be recorded during the simulation as this is the basis for the graph structure. To minimise the effort of software developers having to manually log the required information with every method declaration, techniques from aspect-oriented programming can be used. The intention of aspect-oriented programming is to improve the modularity and maintainability of implemented software by separating *crosscutting concerns* that are typically spread across different places in the source code from the main functions of the program [299]. In particular, crosscutting concerns are implemented in a central place as an *aspect* which prevents later changes having to be made in several places of the source code and thus facilitates their maintainability. During the compilation process, these centrally implemented aspects are generated (*weaved*) to the corresponding places in the source code (*join points*). This ensures that aspects are applied at the right time when the program is executed (see Figure 6.5).

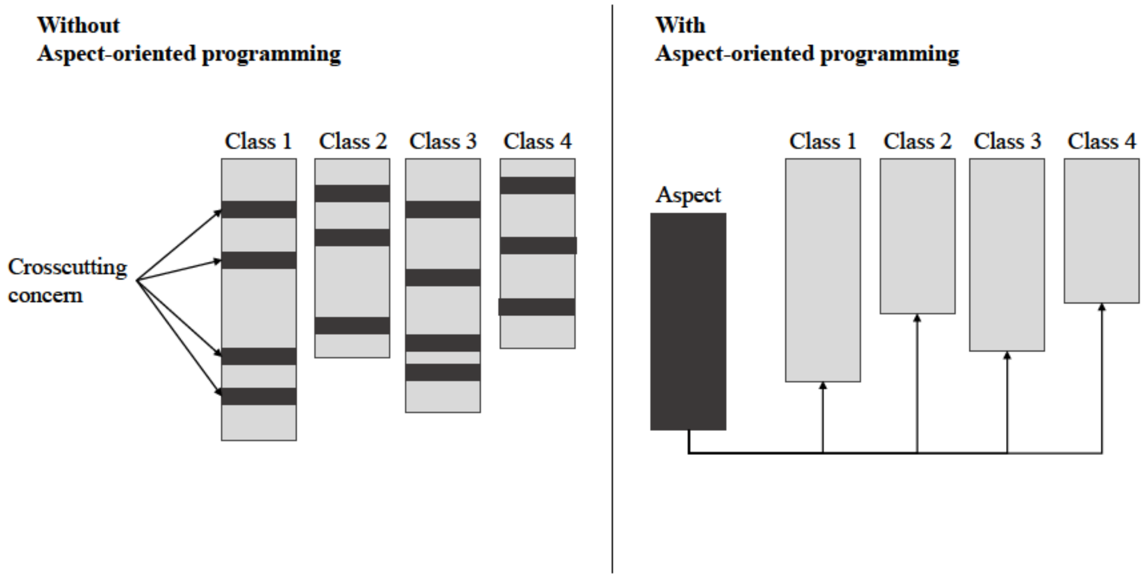

Figure 6.5: Aspect-oriented programming.

In this research, the code for logging the relevant information that is later used to generate the MAMN graphs is implemented as an aspect. For this purpose, two join points have been defined at which the aspect is executed. In particular, relevant information needs to be logged before either a method (`beforeMethod`) is executed or a constructor (`beforeConstructor`) is used to create a new object (e.g. an agent). Given that agent-oriented programming is a specialisation of object-oriented programming [82], relevant information to be collected during methods calls can be specified as follows (see Table 6.1). In addition to this, the logger aspect needs to include a globally incremented counter that serves as the log id and indicates the order of log entries. It is important to note that multi-agent simulations are typically multi-threaded and therefore the order of log entries cannot be taken as a given. To overcome this problem, log entries need to include information on their thread id. This allows the order of the log entries within the same thread to be identified. The process of determining the order of the log entries across different threads will be further dis-

Table 6.1: Relevant information on method calls to be extracted.

Relevant Information	Description
Method name	The name of the current method that is being executed which is relevant for creating activity nodes.
Class name	The name of the class in which the current method that is being executed has been declared to uniquely identify the activity to avoid confusion about methods with the same names in the source code.
Input variables (type and name)	The type and the name of input variables of the current method.
Method type	Information on whether the current method is a constructor to create appropriate edges in the graph.
Caller method name	The name of the previous method from which the current method has been initiated.
Caller class name	The name of class in which the caller method has been declared.
Current thread id	As multi-agent simulations are typically multi-threaded, information is required on the current thread id.

cussed in the second step. Formally, a log entry l that is produced during the simulation for a method call m is collected in a list L. Therefore, $l \in L$ for which applies $l = \{id(l), tId(m), n(m), c(m), n(m_{predecessor}), c(m_{predecessor})\} \cup I_m$. To further specify, $id(l)$ is the assigned id for log l based on the globally incremented counter of the logger aspect and $tId(m)$ is the id of the thread in which m was executed. $n(m)$ and $c(m)$ are the name of the method as well as the name of the associated class in which m was declared. In the same manner, $n(m_{predecessor})$ and $c(m_{predecessor})$ are the method and class name of $m_{predecessor}$, where $m_{predecessor}$ is the preceding method from which m was initiated. Finally, I_m defines the set of input variables $i \in I$ with $i = \{type, name\}$. $type$ contains information on the associated data type for input variable i while $name$ is the assigned variable name. To conclude the activities of the first step, L is serialised and output to a file for further processing.

In the second step, log entries L are aggregated and processed into a linked list structure that reflects the flow of the main activities of the simulation. This step takes place after the computation of the simulation has been completed. For this purpose, log entries are loaded and deserialised from the output file. Log entries $l \in L$ are sorted first by their thread id t_{id} and then by their log id l_{id}. This gives insight into the order of log entries within the same thread t. As described above log entries contain information about a method call m as well as their predecessor $m_{predecessor}$ from which they have been initiated. It can be noted that for the method m_{start} that was used to start the simulation, the resulting log entry contains no predecessor. In this case, the log entry was already marked as the *entrypoint* when it was created. Assuming that threads are typically terminated after completing their task and that new threads are created whenever new tasks need to be processed in parallel, sorting the log entries by their thread id already ensures that methods in the log entries list always refer to a predecessor that is located somewhere earlier in the list. However, it is not guaranteed that the neighbouring item in the list is also the relevant predecessor. As the log entry l of a current method $m_{current}$ contains information about the class $c(m_{predecessor})$ and method $n(m_{predecessor})$ from which it was initiated, the relevant preceding log entry $l(m_{predecessor})$ can be identified by iterating backwards in the list and finding the last entry that matches the referenced predecessor class and method name (i.e. $c(m_{predecessor}) == c(m_{currentIteration})$ && $n(m_{predecessor}) == n(m_{currentIteration})$).

Based on this, the list of sorted log entries L can be processed into a linked list structure that reflects the flow of activities in the simulation F. For this purpose, let there be an activity $a = (l, type, S_a)$ for each $l \in L$. Thus, a is a triple consisting of a log entry l, its $type$, and an ordered list S_a of its successors. The type indicates whether a is executed once or as a loop. S_a is empty at the beginning when a is created from l. With subsequent iterations over L, the activity of the current iteration $a_{current}$ are added to the list S_a of the preceding activity. To reduce the complexity of the flow of activities for later representation as MAMN graphs, activities $a_{current}$ are only added to the list of successors when the class and method name of the latest successor $a_{latest} \in S_a$ does not match the same properties of $a_{current}$. In the case that the properties of $a_{latest} \in S_a$ and $a_{current}$ match, this indicates that the activity $a_{latest} \in S_a$ is being repeated. Therefore, the type of the latest successor $a_{latest} \in S_a$ is changed to being a *loop* activity and $a_{current}$ will not be added to S_a one more time. Successors $s \in S_a$ of activity a thus only contain activities that are directly initiated by a. This concludes the creation of F being a linked list structure to reflect the flow of activities in the simulation (see Algorithm 2).

Finally, in the third step, activities in the linked list structure F are used to generate

Algorithm 2 Algorithm to process log entries into the linked list structure.

Require: List of log entries L

Initialise an empty linked list F

for each log entry $l \in L$ **do**

Initialise an empty list S_a

Create an activity $a = (l, \text{type}, S_a)$

for each log entry $l_{current} \in L$ **do**

if $l_{current}$ refers to l_a as the predecessor **then**

$a_{current} \leftarrow$ activity from $l_{current}$

$a_{latest} \leftarrow$ last activity element in S_a

if class and method name of $a_{latest} \in S_a$ differ from $a_{current}$ **then**

Add $a_{current}$ to S_a

else

Change type of latest successor $a_{latest} \in S_a$ to *loop*

end if

end if

end for

Add a to the end of F

end for

return linked list F

MAMN graphs. In particular, separate graphs are created for different hierarchical levels. Hierarchical levels are derived from the activities and their immediate successors in the linked list structure. The proposed framework allows the end-users to select the level of detail at which they would like to generate and look at MAMN graphs. For this purpose, a breadth-first search is used to iterate through the linked list (see [306]). By default, an MAMN graph is generated for each hierarchical level if the list of successors contains more than one element. Let a be the current activity with the list of immediate successors S_a. Depending on the user-configured minimum number of items in S_a a separate MAMN graph will be generated for a. The overall description of the graph is given by the name of activity a. Each graph contains a start and end node. Further nodes and edges of the graph are derived from elements $s \in S_a$. For each activity $s \in S_a$, an activity node is created. In the case that s contains a list of successors S_s with a number of elements that is greater than the user-configured minimum m, instead a node is created for s being an activity with subgraph. Furthermore, an MAMN variable node element (e.g. primitive, complex or collection variables) is created according to the name and type of each input variable of the activity.

Apart from this, edges of the graph can be derived from activities $s \in S_a$. The start node is connected to the first element of S_a while the last element in S_a points to the end node. Typically the order of activities $s \in S_a$ determines the flow of activities for the graph. Thus, $s \in S_a$ points to the next activity in the successors list. However, there is an exception if s is an activity of the type constructor. In this case, it is first verified whether there is already a variable node for the result of the constructor activity. In the case that there is not already one, a new variable node is created. The constructor activity node is then linked to the resulting variable node. This variable node then refers to the next activity in the successor list. Furthermore, input variables point to the associated activity. By default, a causal relation is created. Specialised edges are based on the information of the node elements involved e.g. loop activities typically result in outgoing for each relations while a constructor activity results in a constructor edge.

Using the proposed method, nodes and edges for the graph can be derived in a fully automated process based on the information extracted from the aspect-oriented logging mechanism. By default, this creates a graph which demonstrates the flow of activities in the simulation as well as the associated input variables for each activity. As an extension, the MAMN notation generally allows mental-level attributes of the agents to be highlighted in the graph structure through dedicated notation elements (mental-level nodes). These mental-level nodes are specialisations of the variable and activity nodes. For the automated process to differentiate mental-level nodes from conventional variable and activity nodes, relevant BDI properties need to be clearly identifiable in the source code. When using a specialised framework such as JADEX for implementing BDI agents in the simulation, the framework uses annotations to flag BDI properties (beliefs, desires, intentions) at the relevant attributes and methods in the classes of the source code [240]. In this case, the automated process for deriving the nodes and edges of the graph structure is able to parse these annotations during the logging process and thus change the type of the affected variable or activity node to the corresponding mental-level node. However, if the simulation was implemented without a specialised framework that uses this type of annotation, highlighting mental-level attributes of agents in the graph requires annotations to be added manually. Based on the experience with the JADEX framework required time and complexity for adding this type of annotation is minimal even when not using the JADEX framework. Note that annotations on BDI properties are not necessarily required if the intention is only to look at the flow of activities and the associated input variables in the simulation. This concludes the process for deriving node and edge elements for the MAMN graph (see Algorithm 3).

Algorithm 3 Algorithm to process the linked list structure into MAMN graphs.

```
Initialise empty set of MAMN graphs G and breadth-first queue Q
Enqueue F to Q
while Q is not empty do
    Dequeue a from Q
    S_a ← immediate successors of a
    if |S_s| > m (user-configured minimum) then
        Create a new MAMN graph G_a
        Create nodes S_start and S_end in G_a
        for each activity s ∈ S_a do
            if |S_s| > m then
                Create a node for s as an activity with subgraph in G_a
            else
                Create an activity node S_s in G_a
            end if
            for each input variable of s do
                if variable node does not exist then
                    Create an MAMN variable node in G_a based on name and type
                end if
                Link variable node to s
            end for
            if s is a constructor activity then
                if resulting variable node does not exist then
                    Create a new variable node for the result of s
                end if
                Link constructor activity node to the resulting variable node
                Link the resulting variable node to the next activity in S_a
            else
                Link s to the next activity in S_a
            end if
        end for
        Link S_start to first element in S_a
        Link last element in S_a to S_end
        Add G_a to G
        Enqueue immediate successors of a to Q
    end if
end while
return Set of MAMN graphs G
```

Although the original intention of MAMN is to demonstrate cause-effect relations of input and output variables, the implementation of the framework has revealed that information on the output variables is particularly difficult to extract. This is due to not every method having a return value. The presented method for extracting the relevant information on cause-effect relations therefore does not fully manage to leverage the expressive possibilites of the MAMN notation. However, retrieved information can still be useful to understand how variables are processed within the simulation. Nodes and edges may then serve as input for graph visualisation. In this research, visualisation has been implemented using the open source library *Graphviz*.[1]

6.2 Demonstration

As proof of concept, the proposed graph-based framework has been applied to extract relevant information and to generate the associated MAMN graphs for two different simulations. In the first case, MAMN graphs are generated for a reduced example of the grocery shopping simulation from Chapter 4.3.1. In the second example, the framework is applied to a published simulation that deals with a commuter scenario in Edinburgh [307].

6.2.1 Generating Cause-effect Graphs for a Reduced Example of the Grocery Shopping Simulation

To demonstrate the proper functioning of the graph-based framework, a reduced example of the grocery shopping simulation has been prepared. In this reduced example, the main phases of the agent life cycle remain the same (see Figure 4.4). However, computation in the simulation has been simplified with static values. For example, when agents are supposed to move in the simulation there are no routes being computed based on routing algorithms applied to geographical map data. Instead, distances are statically assigned to the agents imitating agent movement. The focus of this experi-

[1] See `https://www.graphviz.org/` - (accessed on 11/10/2023)

mentation is on showing the causal relations when processing and passing on variables during the simulation rather than looking at the actual values of these variables (providing additional information towards structural validation as part of computerised model verification rather than operational validation). Thus it is possible to look at smaller example simulations when trying to understand the internal mechanisms of the model. In this section, the proposed framework has been applied to extract relevant information for a simulation with a population of ten agents performing the activity of grocery shopping. This resulted in a log with 99 entries (see Appendix A) that has been processed to first obtain the linked list structure and then produce the visualisation as MAMN graphs.

At the top level, the MAMN graph is rather straightforward (see Figure 6.6). The simulation starts by creating a `SimulationController`. This `SimulationController` initiates an activity to `performSimulation()`. The `performSimulation()` activity is given as input variables the `numberOfAgents` and a collection of `personas` while details of the `performSimulation()` activity are shifted into a subgraph (see Figure 6.7). This is where the actual simulation takes place. The subgraph in Figure 6.7 starts with the creation of the `AgentFactory`. This is a constructor activity, which is why the activity `AgentFactory.init()` points to the resulting variable node `AgentFactory`. Based on this, an agent population is generated in `createAgents()` using the input variables `numberOfAgents` and the collection of `personas`. The agent population can be ac-

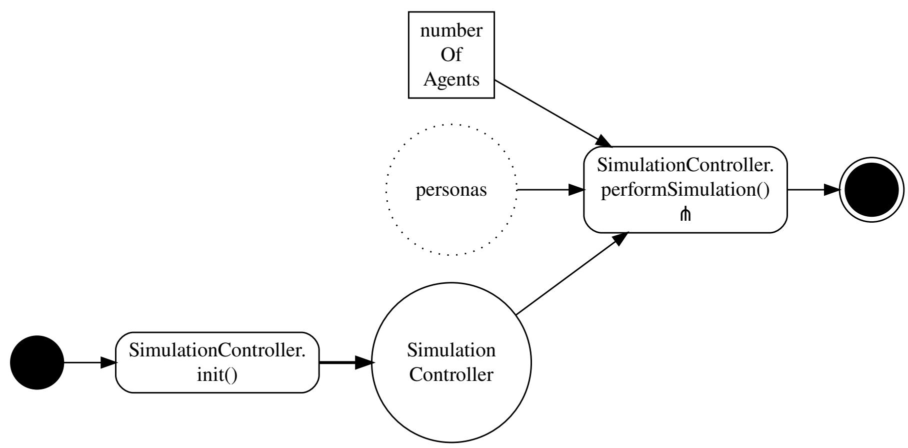

Figure 6.6: Minimal Example - Activities on the top level: Starting the simulation with the two user inputs *number of agents* and the list of *personas*.

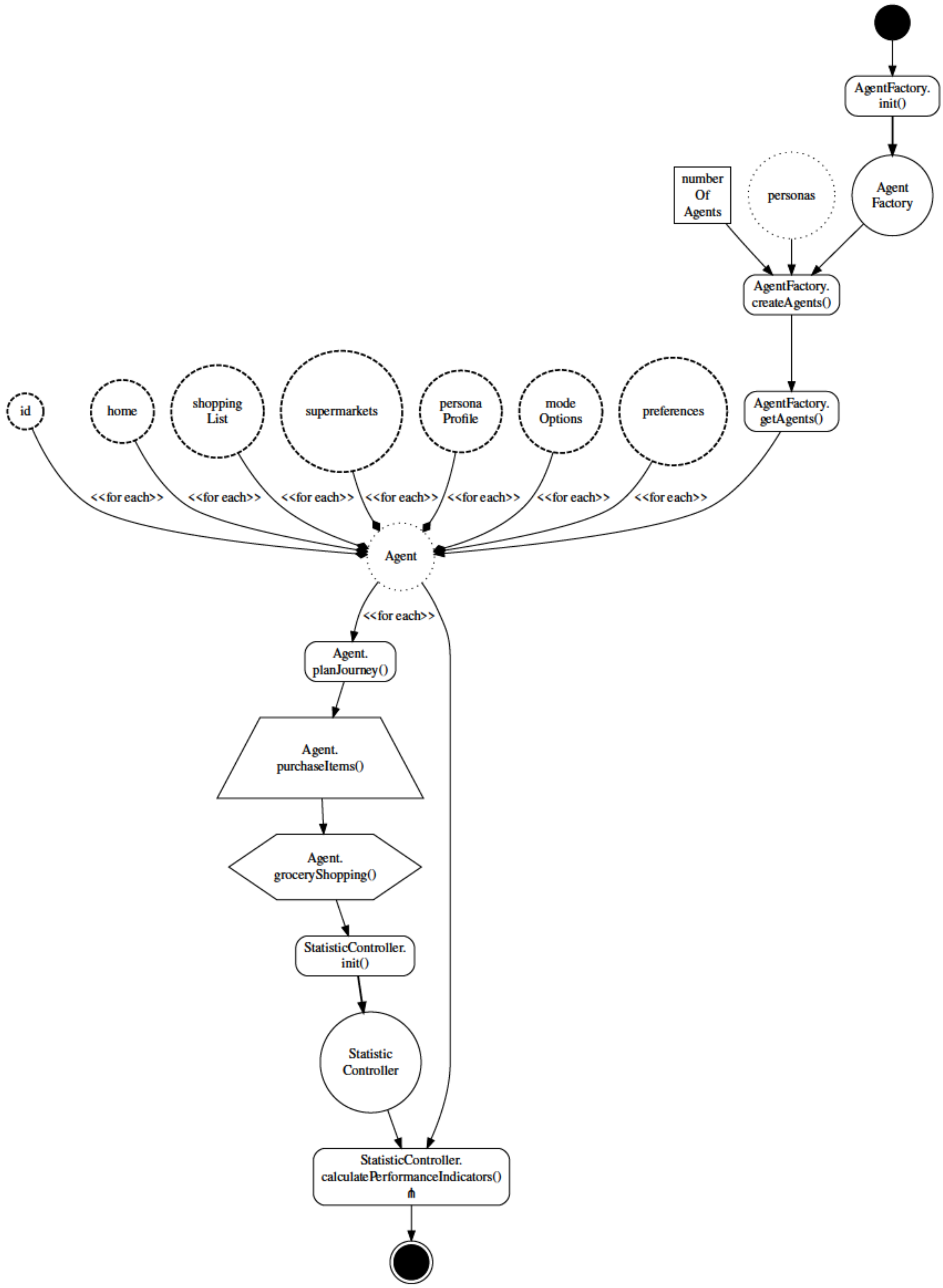

Figure 6.7: Minimal Example - Subgraph for performSimulation(): Illustrating the beliefs of BDI agents and their activities during the simulation.

cessed via the `getAgents()` activity which results in the variable node for the collection of `agents`. As agents and their beliefs have been annotated in the source code to indicate their mental-level properties, the associated belief nodes are linked to the collection of agents node using the contain relation. The outgoing *foreach* relation indicates that for each agent within the collection a `planJourney()` activity is initiated. When the `planjourney()` activity is completed, the agent proceeds with the `purchaseItems()` activity. `purchaseItems()` is indicated as a mental-level desire node that initiates the `groceryShopping()` intention. Once the grocery shopping activity completes, a `StatisticController` is created which then executes the `calculatePerfomance-Indicators()` activity. Again, details of `calculatePerfomanceIndicators()` are shifted to a subgraph (see Figure 6.8). During this process, two performance indicators are computed for global travelling distance as well as global combustion distance. At this point, the example simulation comes to an end.

The systematic approach of the proposed method enables reverse engineering of the internal mechanisms of the simulation at runtime in an automated process and thus may help to ensure completeness and correctness of the obtained graph structure. Obtained graphs in this example provide information on the ordered sequence of activities during the simulation in an easily accessible manner. Inputs and outputs of activities are associated with cause and effect relations, as inputs to activities are used to compute the output variables. The representation of these relations as MAMN graphs gives an overview of the simulation workflow as well as the chaining of preceding activities and variables that have an influence on the computation of output/ result variables. This type of information is an integral part of comprehending the implementation of a simulation model especially when working with source code that was implemented by a different contributor.

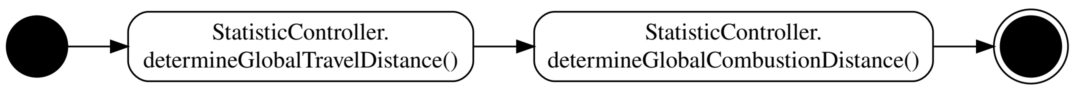

Figure 6.8: Minimal Example - Subgraph for calculatePerformanceIndicators(): Computing the two performance indicators *global travelling distance* and *global combustion distance* at the end of the simulation.

6.2.2 Generating Cause-effect Graphs for a Published Simulation Model on Commuter Traffic

In the second example, MAMN graphs have been generated for a published simulation model (see [307]). This simulation was executed as given and has not been simplified as done with the grocery shopping simulation. The focus of the simulation was to examine the modal choices of individuals when part of the staff has to permanently change workplace e.g. when whole departments are relocated to another office within the city. Given the new circumstances, individuals may have shorter or even longer commutes depending on where they have their residency in the surrounding area. This changes public transport connections as well as the overall costs for individuals to be reconsidered. As a consequence, individuals are likely to change modes of transport which can have different types of effects. For example, permanently relocating a department to another office may cause overall travel distances to increase. This will lead to more employees travelling in private vehicles as they will not be able to continue walking or cycling and thus has a negative impact on the environment. Individuals in this simulation have also been implemented following the BDI model.

This time the proposed framework has been applied to a given scenario with 22 agents that are simulated over 30 days. Extracted information from the simulation at runtime resulted in a total of 80922 log entries. Based on the aggregation mechanisms that take place during the processing of the list of logs into the linked list structure as well as the computation of nodes and edges for the MAMN graphs, it was possible to achieve a compact representation of the simulation. At the top level, the simulation starts with creating an instance of the `SimParams` object that holds input configurations for the simulation (see Figure 6.9).

This is modelled in the graph as an `SimParams.getInstance()` activity that results in a complex variable `SimParams` using a constructor relation. This SimParams variable is subsequently complemented with information on the user configurations (e.g. output file, map data (osmFile) or how many days are to be simulated) using setter functions in the source code. Each set operation is modelled in the MAMN graph using an

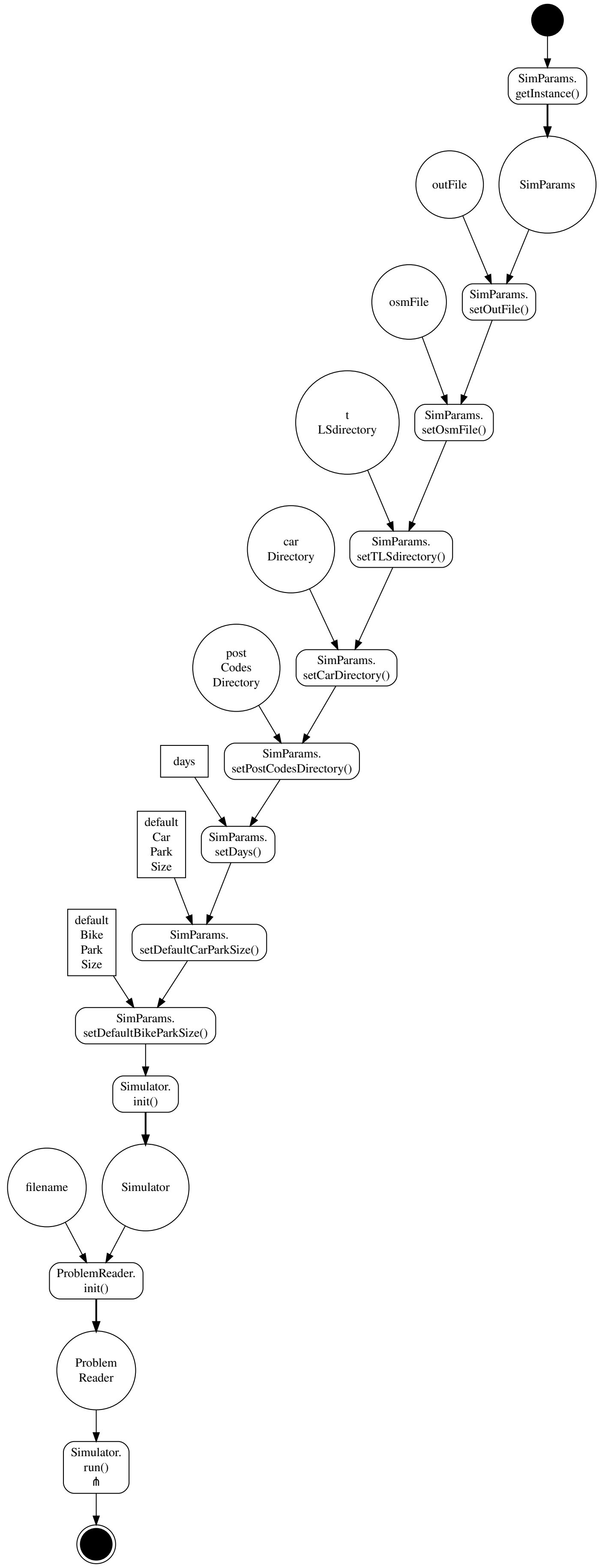

Figure 6.9: Commuters Simulation - Activities on the top level: Processing user inputs.

activity node together with the associated variable node for the input information. When set operations for defining user configurations are completed, a `Simulator` object is created using the same notation elements as already seen. Following this, a `ProblemReader` is created that receives a filename in which data on the problem scenario is defined. The top level concludes with an activity to run the simulation. Analogous to the previous example details of running the simulation are shifted to a subgraph (see Figure 6.10).

In the subgraph, the simulation starts by initiating the `newDay()` activity within the `TransportationManager`. Agents in the simulation are modelled in the collection node `SimpleBDICommuter` and have mental-level belief properties on their patience, home and work location as start and end time of the work day. As the graph illustrates each agent selects a travel option for the current day which is modelled as a mental-level desire property and initiates the intention activity to get the cheapest option in time. Following this, the simulation proceeds with processing simulated information on commuter feedback before writing results and starting the next iteration (the next day) of the simulation. At this level, details on the activities to `selectTravelOptions()` (see Figure 6.11) as well as `writeResults()` (see Figure 6.12) have been shifted to subgraphs. Figure 6.11 on the activity to `selectTravelOptions()` essentially illustrates the creation of the agent journey given two locations as input. In the simulated scenario, these locations refer to the home and work location of the agent. The journey is further specified with information about the start time as well as the direction of travel (home to work or returning home from work) and also holds information about the commuter as well as mode of transport. Apart from this, figure 6.12 on the activity `writeResults()` shows the process of how results in the simulation are output. In this context, it becomes apparent that performance indicators examined in the simulation relate to modal choice as well as other indicators regarding required costs, emissions and travel time of the determined agent journey.

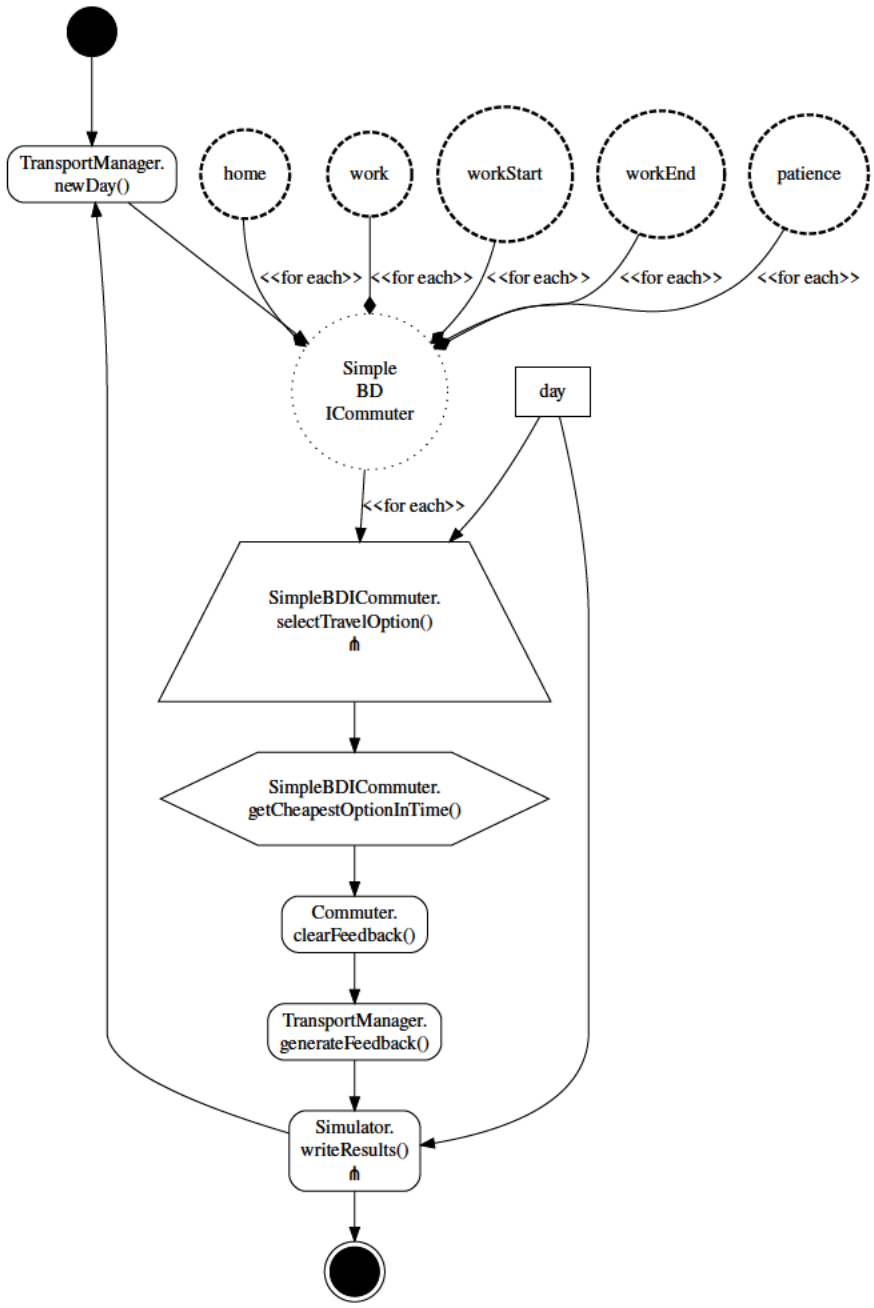

Figure 6.10: Commuters Simulation - Subgraph for Simulator.run(): Illustrating the beliefs of SimpleBDICommuters and their activities during the simulation.

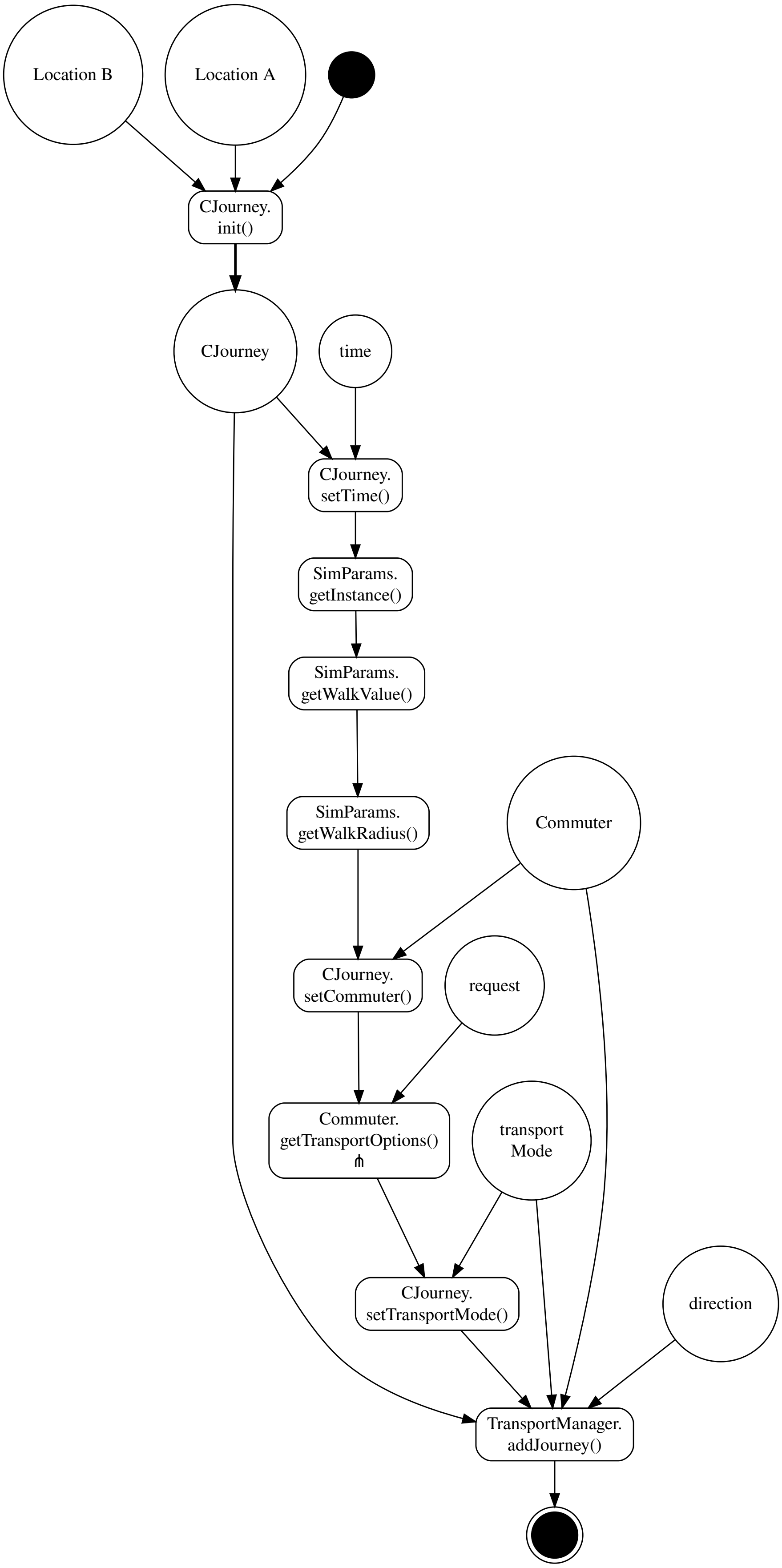

Figure 6.11: Commuters Simulation - Subgraph for Commuter.selectTravelOption(): Variables and activites involved for selecting the travel option of an agent.

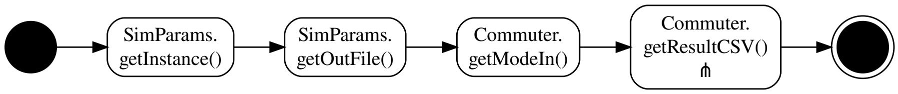

Figure 6.12: Commuters Simulation - Subgraph for Simulator.writeResults().

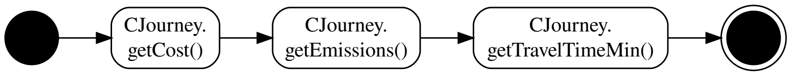

Figure 6.13: Commuters Simulation - Subgraph for Commuter.getResultCSV(): Documenting the performance indicators *costs*, *emissions* and *travel time* at the end of the simulation.

All in all, the generated graphs and visualisations provide a means to gain insights into the internal mechanisms of the simulation without having to manually read the source code. This can be helpful to get a quick overview of the simulation model when working with source code that was implemented by a different contributor. The systematic and automated process for extracting and representing cause-effect relations of the simulation may help to ensure completeness and correctness of the graph structures. The representation of cause-effect relations as MAMN graphs gives an overview of the simulation workflow as well as the chaining of preceding activities and variables that have an influence on the computation of output/ result variables. This provides additional explanatory information about the simulation and thus can be considered as a contribution towards achieving more trust in the results of simulations by illustrating the process of how results are computed. The compact representation of the simulation model as graphs is the result of aggregations during the processing of raw log entries retrieved from the running software. As a consequence, insights provided by the compact representation are bound to the considered level of abstraction. Results presented in this research have demonstrated that reverse engineering the internal mechanisms of agent-based simulations can generally be achieved. In particular, by automatically extracting cause-effect relations using appropriate logging mechanisms that are based on techniques from aspect-oriented programming and representing these relations in graph structures. This concludes the demonstration of the proposed graph framework.

6.3 Summary of Chapter 6

In this chapter, a method has been proposed to extract and formally represent cause-effect relations of agent-based traffic simulations to provide more explanatory information on the internal mechanisms. The basic idea is to capture the cause-effect relations of input and output variables in a graphical model. Cause-effect graphs have already been used to model aspects of simulations, but typically look at relations on a single level of abstraction. However, the multi-level property of agent-based simulations shifts cause-effect relations of the simulation into several levels of details and abstraction. This is why conventional cause-effect graphs need to be extended. For this purpose, section 6.1.1 presents a dedicated profile of UML activity diagrams that is called Multi-Agent Modelling Notation (MAMN). The MAMN graph structure shifts the details of the hierarchical levels into separate subgraphs. Furthermore, UML activity diagrams as an implementation of behavioural diagrams have been extended by structural elements for modelling the information on input and output variables as well as BDI properties of the agents. MAMN graphs are a compact representation of the simulation that helps to obtain an overview of the internal mechanisms.

In section 6.1.2, a method is proposed to automatically extract relevant information on the cause-effect relations of input and output variables and to represent these relations as MAMN graphs. For this purpose, relevant information is logged during the simulation each time a method or constructor is executed. To reduce the manual workload for logging, the proposed method uses techniques from aspect-oriented programming. Extracted information in the list of log entries has been further processed first into a linked list structure and then into the corresponding nodes and edges for the MAMN graphs. It should be noted that the proposed method may be limited when dealing with recursion. However, this can be compensated for by using iterative alternatives instead of recursions. Using the open-source library Graphviz, nodes and edges have been visualised as MAMN graphs. For demonstration purposes, section 6.2 applies the proposed method for extracting relevant information on cause-effect relations and generates graphs to a reduced example of the grocery simulation from chapter 4.3.1 as well as an external simulation on commuter traffic.

Chapter 7

Critical Evaluation

As the number of digital and personalised services increases in the mobility sector, individuals are given more flexibility when it comes to arranging their personal mobility. This leads to changes in the behaviour of individuals and ultimately has an effect on the overall traffic situation. Traffic simulations that are used to study the effectiveness and efficiency of new mobility schemes need to be prepared to deal with the changing requirements. For this purpose, this research has been concerned with assessing the requirements placed on traffic simulators in response to the ongoing developments in mobility. Based on this, the aim of this research has been to develop appropriate methods to cope with these requirements when building simulations for contemporary scenarios of interest. This chapter provides a detailed summary and discussion of the main contributions of this research. The intention is to reflect on the applicability and relevance of the research findings in a broader context. For this purpose, contributions and results are discussed with regard to their innovation, scope and limitations. Research methodology as well as chosen technologies are critically evaluated by highlighting the advantages and drawbacks of the chosen approach and comparing it to possible alternative research paths based on available literature. Furthermore, a discussion is given of the experimentation performed in the scope of this research, reflecting on the design of experiments, obtained results as well as lessons learnt.

7.1 Discussion of Contributions to Knowledge

Placing the individual in the center of attention for simulating current scenarios of interest leads to new requirements when it comes to their implementation. Consequently, there is a need for appropriate methods and tools to facilitate the development of individual-based traffic simulations. Contributions of this work focus on obtaining an overview of the current challenges in mobility as well as the current state of implementation of available traffic simulators to illustrate their scope and limitations. These limitations have been addressed as the subject of this work. Based on this, this research proposes a set of methods to facilitate the development of individual-based traffic simulations. The proposed methods have been implemented as prototypical tools to demonstrate their applicability in a selection of example use cases.

7.1.1 Contribution 1: A Systematic Survey of Available Agent-based Traffic Simulators

To get an overview of the current challenges in mobility as well as the wide spectrum of available traffic simulators, in this research, a systematic survey has been performed based on available publications. Admittedly, there is related work that has also given an overview of available simulators e.g. [76], [80]. However, these publications were published several years ago since when new applications have emerged. Yet the main reason for conducting and publishing another systematic survey was to look at simulators from another perspective with regard to their ability to model individuals and their behaviour. This is where this research argues that due to the current developments in mobility (e.g. the increasing number of digital and personalised services), the role of individuals and their behaviour in traffic simulations needs to be reconsidered. In 2019, which was shortly before the start of this PhD project, [227] concludes in their work that the implementation of individual behaviour in traffic simulation is limited and therefore explicitly calls for more research in this area. According to Google Scholar, the systematic survey that has resulted from this PhD project [4] has been cited over 40 times since its publication in December 2021. This can be an indicator of the relevance of this subject matter.

The survey was conducted based on a keyword search across the three common publication databases: Google Scholar, ACM Digital Library, and IEEE Xplore; on the title, abstract, and the main body of the papers. The first 30 research papers from each database and each keyword were then included in a backward search to identify simulators that are considered related work by the authors of the publications. In addition to this, it would have been possible to include other databases such as Scopus, Web of Science or JSTOR in the search as well as to consider more papers in general for each database and keyword to make the survey more comprehensive. The selected databases were mainly chosen because easiest access was available. The decision to limit the number of papers was mainly driven by the observation that the exploration of additional research papers beyond this threshold did not yield significant new examples of simulators. Continuing the search may have led to marginal increase in survey comprehensiveness. Overall, the approach of this research was based on methodology described in similar surveys (e.g. [103], [104]). It was also very helpful to have feedback from peer reviewers during the publication process. This has allowed to raise the quality of this survey. Nevertheless, it is always possible that specific publications may have been missed. Since the date of publishing this survey, a small number of simulators have been added to this survey over the course of writing this thesis. For example, relevant or new publications from the research community that have appeared on ResearchGate.

The review of simulators was structured based on current areas of interest in mobility [189]. Furthermore, simulators and implemented features to model individual behaviour were structured by time aspect (short-, mid- and long-term behaviour). Apart from the time aspect, other criteria about the simulators could have also been used to structure the survey (e.g. underlying programming language or licensing model), but these might have shifted the focus of the survey. As the focus of this thesis was to look at the implemented features to model individual behaviour, this review structure was found to be particularly suitable.

The systematic survey in this research has shown that there is a broad range of simulators that each focus on different areas of application. As a result, simulators vary in the implemented features for modelling individuals and their behaviour. A common notion that has been found across reviewed simulators was that there are only a few approaches that even consider the modelling of individual preferences as part of agent behaviour [226]–[228]. Thus, there is a need for new methods and tools that enable the implementation of individual preferences as determining factors of individual decisions. All in all, this systematic survey has helped to obtain an overview of the current state of available traffic simulators and to identify gaps and limitations to be further researched based on a systematic and reproducible method.

7.1.2 Contribution 2: A Modelling Framework to Capture Preferences and Knowledge of Individuals

Based on the findings of the systematic survey, this research proposes as the second contribution of this thesis, a framework to capture preferences and knowledge of individuals as determining factors of agent decisions. As mentioned in chapter 2.4.2, modelling agent decisions and their reasons becomes significantly complex in quantitative models when decisions require a broad knowledge of the world [46]. A qualitative approach can help reduce complexity in the modelling while creating rich agent models. For this purpose, the use of semantic technology has been proposed. Leveraging the ideas from the CommonKADS model, agent knowledge has been structured in a hierarchical architecture with different levels of abstraction. In particular, information for domain (layer 1) and inference knowledge (layer 2) is modelled using rules and ontologies. This allows the application of computer-based reasoners to reduce the complexity during the modelling process. In addition to this, task knowledge (layer 3), in which information from the lower levels is brought together to perform decision-making and determine actions, is implemented in the BDI agent. As a result, this approach leads to a clear separation of the implementation of agent knowledge from its operating behaviour. Agent knowledge can thus be easily extended or replaced which allows them to be flexibly reused across different scenarios. However, the application of semantic technology may potentially increase the complexity of the overall system architecture as well as the simulation model. This may lead to longer simulation times as the computation requires more resources to process the detailed modelling of the individual. Furthermore, modelling more details of the individual requires more input data from the real world, but can also lead to new opportunities to evaluate the effects of traffic policies. In particular, in the proposed method, the preferences of agents are used as input arguments to compound utility functions that model their decision-making. As mentioned in chapter 4.3.1, this research thus takes the utilitarian perspective on utility [247]. [248] has demonstrated the use of utility as experienced by individuals with happiness measures. Based on this, utility functions can be used to quantify experienced utility as an indicator for the satisfaction of individuals. Effects of traffic policies can thus be evaluated not only on global system behaviour but also on individuals. This helps to design and identify effective measures by avoiding public opposition which in the past has led to long implementation times.

To demonstrate the applicability of the proposed method, simulation has been given for two example scenarios. The focus of the experiments was not to present a validated simulation model but to show how the proposed methods can be implemented as well as the benefits and limitations. In particular, during the implementation of the example use cases, possible information sources were explored in order to collect relevant input data for the detailed modelling of individuals. With regard to the experimental design, the simulations were investigated based on exemplary research questions. The intention of these experiments was to create examples of implementation and to provide proof of concept. For the experiments the size of the population was derived from the objects under examination. For these example use cases, calculations were completed in approximately three hours per simulation. This was acceptable for the purpose of the experiments. Larger simulation settings may require code optimisation or high performance computing machinery. Furthermore, it should be noted that the simulations in this research were deterministic when keeping the same agent population. This simplified analysis and proof of concept, making comparison of simulations easier. Depending on the research objective in real-world scenarios, simulations may require a more probabilistic approach.

In the first scenario, the mobility of individuals was simulated during their grocery shopping. This scenario was particularly suitable for demonstrating how complex agent decisions can be modelled that require a broad knowledge of the world when using the proposed method. Travel decisions in this scenario required not only knowledge about the transportation domain but also about food and supermarkets. The variety of available food items as well as the information about which items are offered in which type of supermarkets can be easily modelled using the ontologies. Trying to implement this type of world knowledge in traditional source code would be significantly more complex e.g. by manually implementing classes with attributes and then hard coding the information for instances rather than defining concepts and rules in the ontology to make use of inference mechanisms.

In the second example, simulation was given for a scenario in which individuals travel to a music concert. The main purpose of this example was to demonstrate how implementations of the same agents can be adapted and flexibly reused across different scenarios. Therefore, the knowledge of agents was altered as domain knowledge about food and supermarkets was irrelevant for the scenario. Instead, agent decisions in this scenario focused on the selection of mode of transport, with ridesharing being introduced as a new mode option.

Reflecting on the conducted experiments and the proposed framework, modelling and simulating more details of the individuals is important to appropriately reflect current developments in mobility as well as to address the limitations of evaluating the effects of policies not only on global system behaviour but also on individuals. However, this approach accounts for extra effort in collecting the necessary input data as well as the modelling process itself. While the use of semantic technologies can reduce the modelling complexity to a certain extent, there will always be an additional overhead when increasing the level of detail in the modelling of individuals and their behaviour.

Furthermore, the proposed method currently displays a combination of qualitative and quantitative modelling techniques. While the world knowledge of the agents is modelled qualitatively in the ontologies, preferences at some point are quantified using a Likert scale when they are used as input arguments for utility-based decision-making. This approach was chosen in view that it is easier to evaluate the effects of traffic policies with a utilitarian perspective on utility. In an alternative research path, it might have been interesting to work out how to better exploit the qualitative BDI mechanisms when incorporating the information of preferences and domain knowledge in the decision-making. The current approach uses the mental levels based on the BDI model mainly for structuring information within the agent.

7.1.3 Contribution 3: A Modelling Notation to Capture Cause-Effect Relations in Multi-agent Simulations

As traffic simulations are becoming more complex when modelling more details on individuals and their behaviour, there is a need for appropriate tools to generate more insights into the internal mechanisms of the simulation. As discussed in chapter 5, the required view on agent-based simulations that is relevant for their structural validation is an ongoing challenge due to inputs of the simulation being modelled on the individual level whereas results on system behaviour are typically examined using performance indicators from the global perspective. Current methods for structural validation are mainly based on face validation during which domain experts manually read source code in structured walkthroughs together with the involved software developers. A graphical representation of cause-effect relations can provide additional explanatory information about the internal mechanisms of the simulation but requires a graph structure that is able to capture the multi-level property of agent-based traffic simulations. Therefore, as a third contribution, this research proposes a graphical notation (MAMN) that is able to capture the cause-effect relations in agent-based traffic simulations.

MAMN has been created as an extension of the metamodel of UML activity diagrams. The proposed graph structure thus is a dedicated profile for agent-based simulations. As an alternative to the metamodel of UML activity diagrams, it would have also been possible to base the new graph structure on BPMN [308]. However, as the intention was to extend information on the program flow with structural information about the input and output variables, UML already contains modelling elements for structural aspects which can be reused. Basing the graph structure on a reference metamodel ensures compatibility with existing standards which makes it easier for end users to adopt the new notation. Furthermore, this approach helps to maintain consistency and coherence in the notation and facilitates its ability to be adapted or extended for future use cases.

The presented MAMN graph currently focuses on showing the input and output relations (cause and effect) of processed variables in a simulation over the different levels of hierarchy. However, there is still potential to extend the graph with specialisations for other types of agent architectures beyond the scope of mental-level models. Furthermore, it is currently possible to model the activities of groups of agents having to get input information for their decision-making based on interactions with other groups of agents. However, there may be scenarios in which the interactions of agents can be difficult to abstract and aggregate which can lead to graphs becoming complex.

All in all, it was possible in the scope of this research to identify a limitation in existing approaches to model cause-effect relationships between input and output variables due to the multi-level property of agent-based simulations. For this purpose, the MAMN graph structure has been proposed. MAMN provides a means to model the hierarchical structure of cause-effect relationships in agent-based simulations from input parameters at the individual level to performance indicators at the global system level. This is achieved by moving details of the functional relationships to a sub-level as a separate graph.

7.1.4 Contribution 4: A Framework to Extract Relevant Information and Visualise Cause-effect Graphs

This research therefore proposes, as the final contribution of this thesis, a framework to automatically extract relevant information on the cause-effect relations of input and output variables from the simulation. Furthermore, the framework also includes transformation procedures to process and represent extracted information as MAMN graphs. In particular, relevant information is extracted to produce a detailed log of computed activities from the simulation at runtime. This mechanism is based on techniques from aspect-oriented programming which minimises the effort of software developers having to manually log the required information. The automated process may help to ensure completeness and consistency of the extracted information on simulation activities. Collecting this information is an overhead but it is usually possible to demonstrate the internal mechanisms of the model based on example simulations with smaller agent populations.

Extracted information in the list of log entries is then sorted first by their thread id and then by their log id. As a result, the order of log entries within the same thread can be derived as a first step. Furthermore, leveraging information from the method call stack allows log entries to be processed into a linked list structure from which nodes and edges can be derived for the MAMN graph. Although the original intention of MAMN is to demonstrate cause-effect relations of input and output variables, the implementation of the framework has revealed that information on the output variables is particularly difficult to extract. This is due to not every method having a return value. In contrast, input variables can be retrieved from the method declaration. All in all, obtained visualisations are still useful for understanding how variables are processed within the simulation. For visualisation purposes, derived nodes and edges may serve as input to common graph visualisation frameworks. In this research, visualisation is based on the open-source library GraphViz. However, there is a variety of alternatives that could have been used for this purpose e.g. [309], [310]. As proof of concept, relevant information about cause-effect relations has been extracted and visualised as MAMN graphs for two example simulations. The first simulation provided a reduced example of the grocery shopping scenario from chapter 4.3.1. In the second example, application of the proposed method has been demonstrated for published simulation.

For reasons of completeness, it would probably be useful to further look into the matter of extracting information of output variables as well as testing the proposed method in a real-world setting with software engineers and domain experts. It should be noted that in this part of the thesis, the focus is to automatically extract and formally represent cause-effect relations in a graph structure. The primary intention is to improve the explanatory capabilities of agent-based traffic simulations and not to present a comprehensive method for structural validation. Nonetheless, the obtained graphical representation of cause-effect relations can serve as additional information for domain experts to get an overview of the system and thus can be considered a contribution towards improving structural validation based on face validation. A more comprehensive contribution to the verification and validation of agent-based simulations yields the potential for another full thesis project. Based on this, experiments for the proof of concept have been limited to the given examples as more extensive work on this matter would have shifted the focus of this thesis and thus was considered out of scope.

7.2 Current State of Implementation

In the scope of this thesis, proposed theories and methods have been implemented as prototypical tools. The intention was to give examples of implementation that can be used for further academic research purposes. Researchers from the community can use these tools to get first impressions of how the integration of components from the different areas can be achieved. The technology stack used in the experiments of this research is only one way of implementing the proposed methods. Depending on the given situation, there are various other options for implementing the presented ideas in other research projects e.g. researchers that are already working with a simulator that is based on a different programming language may choose a different technology stack for the implementation of proposed concepts.

The latest version of the AGADE Traffic simulator is the example implementation of the proposed concepts from chapter 4 for modelling individuals and preferences and was used for the simulation of the example use cases. For this purpose, BDI agents that are implemented using the JADEX framework have been extended to access world knowledge that is modelled based on semantic technology. Semantic components have been implemented using OWL ontologies and SWRL rules for which relevant APIs (the OWL

API and the SWRL API) have been used to establish the communication between agents and the semantic technology. Implementations of the proposed methods have proven that they can work based on the given example use cases. Examples of implementation can be extended and modified for research on future use cases as well as to further improve and refine methods for working with individual-based simulations. It should be noted, that the focus of the examples of implementation has been directed towards their functionality. Although best practices in software development were followed to the best of the authors' abilities, issues that were not the focus of this thesis were only considered in a secondary manner. For example, there is certainly potential for optimisation in terms of performance when prototypical tools are used for working on real-world use cases that deal with more complex simulation models. This applies not only to the implementation of the AGADE Traffic simulator but also to the framework for extracting and visualising the information of cause-effect relations in agent-based simulations.

7.3 Summary of Chapter 7

This chapter reflects on the overall methodology, contributions as well as the current state of implementation of this research. For this purpose, section 7.1 gives a critical discussion of the benefits and limitations of contributions to knowledge as well as possible alternative research paths. The focus of these contributions is on finding appropriate techniques to improve the model development process when placing individuals at the center of attention in traffic simulations. Modelling knowledge and preferences of individuals as determining factors of agent decisions has been limited in previous work but the use of semantic technologies makes it possible to efficiently supply agents with a broad knowledge of the world. Processing the detailed information of individuals with semantic technology naturally is an additional overhead. However, this approach also allows for the application of intelligent reasoning mechanisms that help to reduce the additional overhead during the modelling process. Apart from this, this research also presented a method to extract relevant information on cause-effect relations from a simulation at runtime and to represent it as MAMN graphs to provide more insight into the internal mechanism of the simulation. Experimentation in this research showed that it is generally possible to automatically obtain the relevant information from the simulation and to process it into a graph structure. Finally, in section 7.2 the current state of implementation has been illustrated. The focus of implementations was on giving examples for the proposed methods. There is certainly potential for optimisation when it comes to aspects such as performance.

Chapter 8

Conclusion and Future Research

This chapter summarises the problem statement and main findings of this thesis and gives a discussion of the degree to which contributions to this research have answered the research questions formulated in Chapter 1.4. Furthermore, general conclusions are drawn on the main propositions of this thesis. The chapter closes by giving suggestions for further research.

8.1 Conclusions on Modelling Individuals for Simulating Contemporary Mobility Scenarios

This thesis looked at the new requirements placed on traffic simulations that are caused by the increasing flexibility in the personal mobility of individuals. In particular, digital connectivity has improved the access of individuals to real-time information as well as led to a growing portfolio of mobility services. This increases the complexity of decision factors but allows individuals to be more spontaneous about their travel decisions. Based on this, the problem statement of this thesis argued that the modelling of individuals needs to be emphasised to align traffic simulations with the new requirements for studying contemporary scenarios in mobility. For this purpose, findings of this thesis to answer the research questions formulated in Chapter 1.4 can be summarised as follows.

8.1.1 Research Question 1

What are the main deficiencies in the modelling of individuals and their behaviour in existing agent-based traffic simulators?

To answer this research question, a systematic survey has been performed to get an overview of available traffic simulators. Literature shows that there is a wide range of traffic simulators that each focus on different aspects of the transportation system. For example, simulated scenarios differ in the timescale in which they are examined (short-, mid- and long-term scenarios). When simulating long-term scenarios such as land use, aspects of individual behaviour refer to decisions about workplace and residency while individual behaviour in mid-term scenarios rather focuses on mode or route choice. Short-term scenarios look at individual behaviour on a micro-scale and thus look at aspects such as lane changing, acceleration or braking behaviour. Reviewing implemented features of simulators for modelling individual behaviour therefore needs to be linked to the scope of application. In this research, simulators have been reviewed with regard to the three areas of application: 1. *Resource Utilisation*, 2. *Digital Connectivity* and 3. *New Forms of Mobility*. Current developments in mobility such as the increasing number of digital and personalized services have led to individuals being more flexible in their decisions on personal mobility. These changes are particularly relevant to modelling the short- and mid-term behaviour of individuals.

With regard to this, available simulators have focused on simulating traffic as the primary subject and thus leave scenario-specific aspects to the responsibility of end-users. Initially, a focus on traffic-related modelling aspects appears obvious as platform developers cannot anticipate the full range of scenarios for which their simulators will eventually be used. Following the same line of reasoning, developers need to assume that the simulators will eventually be customised to fit specific research purposes. It is therefore desirable that common and foreseeable modifications are supported by suitable structures and programming interfaces. With regard to the modelling of individual behaviour, a common requirement is to align traveller decisions with the context of the simulation. In particular, traveller decisions that are based on individual preferences and personal objectives typically differ depending on the simulated scenario. Furthermore, the simulated scenario also changes the perception of individual preferences and thus leads to changes in individual behaviour. For example, time/punctuality which can be a criterion for mode selection has a different value when commuting to work as compared to a social visit. However, the review has shown that there are only a few approaches that even consider the modelling of individual preferences as part of agent behaviour. Thus, in the current state of implementation, there is a lack of concepts to capture these preferences and objectives as determining factors of individual decisions. This hampers the customisation of available simulators to simulate current topics in mobility. Research that elaborates on the modelling of these aspects can help to address this problem.

8.1.2 Research Question 2

How can the knowledge of individuals be modelled to capture their preferences and personal objectives as determining factors of decisions in mobility scenarios?

The decisions of individuals in traffic are influenced by numerous aspects and therefore agents require a broad knowledge of the world. However, the representation of decisions and their reasons becomes significantly complex in quantitative models when decisions require a broad knowledge of the world. Qualitative modelling can help reduce complexity while creating rich agent models. For this purpose, semantic technology can be used. In particular, ontologies are a common and generally accepted instrument from semantic technology to model structured knowledge bases. They can be extended with a set of inference rules that allow the use of computer-based reasoners. The use of computer-based reasoners enables the inference of implicit information in the knowledge base and thus reduces complexity during the modelling process. This combination has proved to be highly useful in practical applications. Based on this, agent knowledge can be shifted into separate ontologies while basic operations of the agents remain part of traditional agent programming. This achieves a clear separation of concerns which makes it possible to look into the field of knowledge engineering to structure agent knowledge. In this research, the traditional BDI agent has been extended with a qualitative model of world knowledge based on the three layers of CommonKADS. The lowest layer contains information on domain knowledge and abstracts common concepts from the travel domain from activity knowledge. As a result, activity knowledge can be flexibly extended or replaced depending on the modelled scenario. In the second layer, this domain knowledge is extended by person-related concepts that describe the attributes of the agent. In particular, census properties from this ontology serve as input to rules that compute agent preferences based on survey data. Information from the first and second layers is used for agent decision-making which is implemented using BDI agents in the third layer. The proposed method thus allows for detailed modelling of the knowledge and preferences of agents to be used as determining factors of individual decisions. In doing so, the additional overhead caused by the detailed modelling of individuals has been minimised through the use of established reasoning mechanisms while the reusability of agents across different scenarios has been improved by providing appropriate customisation options.

8.1.3 Research Question 3

How can relevant cause-effect relations in agent-based traffic simulations be automatically extracted and formally represented?

Placing the individual in the center of attention in agent-based traffic simulations increases the complexity of simulation models due to the modelled level of detail. It is therefore important to develop methods and tools for extending the explanatory capability of agent-based simulations, ensuring that the ability to understand the internal mechanisms of the simulation is not further compromised. Explanations can be given through different presentations. Providing explanations using graphical models achieves a compact representation of the simulation model. For this purpose, computational processes from input to output parameters have been captured as cause-effect graphs. In this context, there has been a particular challenge in modelling the relations between input and variables when using conventional cause-effect graphs as these typically focus on modelling relations at the same level of abstraction. However, system behaviour in multi-agent models is the emergent result of the behavioural patterns of a large set of individuals which changes the paradigm from chained causal relations to several levels of detail and abstraction. Thus, in this research, conventional cause-effect graphs have been extended to capture the hierarchical structure of cause-effect relations.

Graph structures can be utilised in a bi-directional process to either transfer a theoretical simulation model into a concrete implementation as an executable piece of code (*forward engineering*) or to represent information from a given implementation (*backward/ reverse engineering*) which can be used to increase transparency and explainability of a system. As the focus of this thesis lies on backward/ reverse engineering, the intention was to automatically generate cause-effect graphs from the source code of the simulation. For this purpose, a logging mechanism has been implemented in the scope of this research, based on techniques from aspect-oriented programming. Code snippets for logging are implemented as separate aspects and automatically injected into the relevant places in the source code during the compilation process. Based on this, relevant information can be logged on every method and constructor invocation. This approach minimises the manual overhead for logging and helps to ensure consistency as well as completeness in the extracted information. Using the information extracted in the log, activities of the simulation can be transformed into a linked list structure based on the flow of the simulation. This linked list structure can then be used to determine the nodes and edges for the cause-effect graphs. Finally, the obtained nodes and edges may then serve as input to visualisation frameworks.

8.1.4 Synopsis of Main Propositions

Traffic simulations that are used for studying contemporary scenarios in mobility require a more detailed modelling of individuals and their behaviour due to the increasing flexibility in personal mobility that has resulted from individuals having better access to real-time information as well as the growing portfolio of mobility services. This requires knowledge and preferences of individuals to be modelled as determining factors of agent decisions. The use of semantic technology ensures knowledge of individuals can be structured in a form that can be easily managed, allowing agents to be reused across different scenarios. Furthermore, this approach enables the application of computer-based reasoners to reduce the complexity during the modelling process. However, detailed modelling of individuals also increases the overall complexity of the simulation. To ensure that the increasing complexity of the model does not further compromise the ability to understand the internal mechanisms of the simulation, agent-based simulations need to be improved with regard to their explanatory capabilities. This can be achieved by automatically extracting and formally representing cause-effect relations of the simulation as graphs. To capture the hierarchical structure of cause-effect relations in agent-based simulations, the proposed MAMN graph structure can be used. Furthermore, the process for extracting and formally representing cause-effect relations as MAMN graphs can be fully automated using appropriate logging mechanisms that are based on techniques from aspect-oriented programming.

8.2 Suggestions for Future Research

Building on the contributions of this thesis, future research may address the current limitations of the presented methods or leverage the findings of this thesis to build new tools that further advance the development of individual-based traffic simulations. For example, examples of implementations for the proposed methods that have been provided in this thesis can be used for academic research purposes but probably require optimisation when using them for real-world use cases that deal with larger agent populations. Future work may consider creating tools that facilitate the integration of these techniques into existing traffic simulation frameworks to encourage broader utilisation by researchers.

Apart from this, one aspect worth considering is to leverage the analytical capabilities of simulations that have become possible only when elaborating on the detailed modelling of individuals. As discussed in this thesis, current research on traffic simulation has primarily studied the effects of policies by measuring social benefit and thus does not sufficiently consider effects on individuals. The detailed modelling of individuals, in particular, their personal attitude that is represented using utility functions and personal preferences makes it possible to measure the effects of traffic policies not only on global system behaviour but also on individuals. Experimental results in this thesis have only given a flavour of measuring the sentiment of individuals using experienced utility. However, for rigorously evaluating the effects of traffic policies on individuals, a more comprehensive approach is likely needed. A systematic approach to the evaluation of sentiment for groups of individuals can be significantly relevant, defining appropriate performance indicators and meaningful visualisations that can help policymakers with their decision-making. Relevant performance indicators and visualisations should be determined through a survey that looks at the requirements of policymakers and traffic planners for different types of scenarios.

Another aspect for further research may refer to the MAMN graph structure which currently focuses on multi-agent simulations that model decision-making based on the BDI model. The notation could be extended to cover other types of decision-making models, improving the informative value of the graphs across a broader range of simulations. As discussed in this thesis, the proposed method for automatically extracting cause-effect relations and representing them as MAMN graphs allows implementations of simulation models to be reverse engineered. This provides information about the internal mechanisms of the simulation which can be useful for the validation and verification of agent-based simulations. While the course of action for replicate and predictive validation is well understood, assuming that appropriate validation data can be obtained, there are still considerable challenges in the structural validation of agent-based simulations which examines whether results are computed through a plausible process. In this context, the aim is not to validate the simulation results against real-world data but to make sure that the implementation of the model (computerised model) works as it was originally intended (conceptual model). The proposed method therefore holds significant potential to improve the structural validation of agent-based simulations but the usefulness remains to be demonstrated. Based on this, future research should involve generating MAMN graphs for different simulations and performing structural validation by conducting field experiments together with domain experts. An empirical study of this nature helps to evaluate the usefulness of the proposed methods in real-world use cases and to further improve them to meet the requirements for a comprehensive structural validation of agent-based simulations. Improving the automated process for the extraction and graphical representation of the cause-effect relations may provide more information on the output variables which in the current implementation remains a limitation. Furthermore, future work may look into adding information on the weighted influence of variables to the graph, for example by combining the proposed method with sensitivity analysis over multiple runs.

References

[1] J. Nguyen, S. Powers, N. Urquhart, D. Eckerle, T. Farrenkopf and M. Guckert, 'Extending AGADE traffic to simulate auctions in shared mobility services', in *Proceedings of the 37th ECMS International Conference on Modelling and Simulation*, ECMS, 2023, pp. 201–207. DOI: 10.7148/2023-0201.

[2] J. Nguyen, S. Powers, N. Urquhart, T. Farrenkopf and M. Guckert, 'Multi-agent modelling notation (MAMN): A multi-layered graphical modelling notation for agent-based simulations', in *PRIMA 2022: Principles and Practice of Multi-Agent Systems*, Springer, 2022, pp. 640–649. DOI: 10.1007/978-3-031-21203-1_42.

[3] J. Nguyen, S. Powers, N. Urquhart, T. Farrenkopf and M. Guckert, 'Modelling the impact of individual preferences on traffic policies', *SN Computer Science*, vol. 3, no. 5, pp. 1–13, 2022. DOI: 10.1007/s42979-022-01253-3.

[4] J. Nguyen, S. Powers, N. Urquhart, T. Farrenkopf and M. Guckert, 'An overview of agent-based traffic simulators', *Transportation Research Interdisciplinary Perspectives*, vol. 12, 2021, ISSN: 2590-1982. DOI: 10.1016/j.trip.2021.100486.

[5] J. Nguyen, S. Powers, N. Urquhart, T. Farrenkopf and M. Guckert, 'Using AGADE traffic to analyse purpose-driven travel behaviour', in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, Springer, 2021, pp. 363–366. DOI: 10.1007/978-3-030-85739-4_33.

[6] J. Nguyen, S. Powers, N. Urquhart, T. Farrenkopf and M. Guckert, 'Modelling individual preferences to study and predict effects of traffic policies', in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, Springer, 2021, pp. 163–175. DOI: 10.1007/978-3-030-85739-4_14.

[7] J. Nguyen, S. Powers, N. Urquhart, T. Farrenkopf and M. Guckert, 'Using semantic technology to model persona for adaptable agents.', in *Proceedings of the 35th ECMS International Conference on Modelling and Simulation*, ECMS, 2021, pp. 172–178. DOI: 10.7148/2021-0172.

[8] S. Robinson, 'A tutorial on simulation conceptual modeling', in *2017 Winter Simulation Conference (WSC)*, IEEE, Dec. 2017, pp. 565–579. DOI: 10.1109/wsc.2017.8247815.

[9] R. Sargent, 'Verification and validation of simulation models: An advanced tutorial', in *2020 Winter Simulation Conference (WSC)*, IEEE, Dec. 2020, pp. 16–29. DOI: 10.1109/wsc48552.2020.9384052.

[10] H. Schaub, 'Simulation als entscheidungshilfe: Systemisches denken als werkzeug zur beherrschung von komplexität', *Entscheiden in kritischen Situationen*, 2003.

[11] Visual Paradigm, *UML - behavioral diagram vs structural diagram*, accessed on 2019-11-28, 2022. [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/behavior-vs-structural-diagram/.

[12] S. Amor, M. Ali and F. Gargouri, 'Verification of the consistency between use case and activity diagrams - a step towards validation of user requirements', in *Proceedings of the 13th International Conference on Enterprise Information Systems - Volume 3: ICEIS*, INSTICC, SciTePress, 2011, pp. 396–399, ISBN: 978-989-8425-55-3. DOI: 10.5220/0003505503960399.

[13] Object Management Group, 'OMG unified modeling language (OMG UML)', Object Management Group, Tech. Rep., 2017.

[14] B. Égert, T. Koźluk and D. Sutherland, *Infrastructure and growth*, Mar. 2009. DOI: 10.1787/225682848268. [Online]. Available: https://www.oecd-ilibrary.org/content/paper/225682848268.

[15] T. Zech, *Stadt und land: Eine beziehungsgeschichte*, accessed on 2023-10-06, 2018. [Online]. Available: https://www.deutschland.de/de/topic/leben/stadt-und-land-fakten-zu-urbanisierung-und-landflucht.

[16] H. Dennig, A. Burri, M. Hoppe and T. Sauter-Servaes, 'Bicar: Sharing mobility for the last mile', in *European Battery, Hybrid, Fuel Cell Electric Vehicle Congress, Geneva, 14-16 March 2017*, Mar. 2017.

[17] V. Llorent-Bedmar, V. Cobano-Delgado and M. Navarro-Granados, 'The rural exodus of young people from empty spain. socio-educational aspects', *Journal of Rural Studies*, vol. 82, pp. 303–314, Feb. 2021. DOI: 10.1016/j.jrurstud.2021.01.014.

[18] Frankfurter Allgemeine Zeitung, *Immer mehr menschen pendeln nach frankfurt*, https://www.faz.net/aktuell/rhein-main/zahl-der-pendler-in-frankfurt-gestiegen-16049585.html, accessed on 2023-01-10, 2019.

[19] A. Loder, L. Ambühl, M. Menendez and K. Axhausen, 'Understanding traffic capacity of urban networks', *Scientific reports*, vol. 9, no. 1, p. 16 283, Nov. 2019, ISSN: 2045-2322. DOI: 10.1038/s41598-019-51539-5.

[20] Y. M. Räth, M. Balac, S. Hörl and K. W. Axhausen, 'Assessing service characteristics of an automated transit on-demand service', *Journal of Urban Mobility*, vol. 3, p. 100 038, 2023, ISSN: 2667-0917. DOI: https://doi.org/10.1016/j.urbmob.2022.100038. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667091722000267.

[21] Y. Hu, C. Yang and K. Axhausen, 'Multi-modal travel simulation and travel behavior analysis: Case study in shanghai', en, *Arbeitsberichte Verkehrs- und Raumplanung*, vol. 1836, 2023. DOI: https://doi.org/10.3929/ethz-b-000625085.

[22] T. Truong, 'Impacts of digitalisation on travel behaviour in hanoi city centre', in *AIP Conference Proceedings*, vol. 2428, AIP Publishing, 2021. DOI: `10.1063/5.0070725`.

[23] T. Kuhnimhof, R. Buehler, M. Wirtz and D. Kalinowska, 'Travel trends among young adults in germany: Increasing multimodality and declining car use for men', *Journal of Transport Geography*, vol. 24, pp. 443–450, Sep. 2012. DOI: `10.1016/j.jtrangeo.2012.04.018`.

[24] K. Erol, R. Levy and J. Wentworth, 'Application of agent technology to traffic simulation', in *Proceedings of Complex Systems, Intelligent Systems and Interfaces*, 1998.

[25] G. Grilli and J. Curtis, 'Encouraging pro-environmental behaviours: A review of methods and approaches', *Renewable and Sustainable Energy Reviews*, vol. 135, p. 110 039, Jan. 2021, ISSN: 1364-0321. DOI: `10.1016/j.rser.2020.110039`.

[26] M. Nicholas, 'Estimating electric vehicle charging infrastructure costs across major US metropolitan areas', *International Council on Clean Transportation*, vol. 14, no. 11, 2019.

[27] R. Hotten, *BMW and daimler invest 1bn in new car venture*, access on 2023-01-11, 2019. [Online]. Available: `https://www.bbc.com/news/business-47332805`.

[28] E. B. Lieberman, 'Brief history of traffic simulation', *Traffic and Transportation Simulation*, vol. 17, 2014.

[29] B. P. Zeigler, A. Muzy and E. Kofman, *Theory of modeling and simulation: discrete event & iterative system computational foundations*. Academic press, 2018, ISBN: 978-0128133705.

[30] A. Law and W. Kelton, *Simulation modeling and analysis*. Mcgraw-hill New York, 2015, vol. 5.

[31] W. P. Kowalk, *System, Modell, Programm: Vom GOTO zur objektorientierten Programmierung*. Spektrum Akad. Verlag, 1996, ISBN: 978-3827400628.

[32] M. Pidd, 'Why modelling and model use matter', *Journal of the Operational Research Society*, vol. 61, no. 1, pp. 14–24, Jan. 2010. DOI: 10.1057/jors.2009.141.

[33] B. Edmonds and S. Moss, 'From KISS to KIDS – an 'anti-simplistic' modelling approach', in *International workshop on multi-agent systems and agent-based simulation*, Springer, 2004, pp. 130–144, ISBN: 9783540322436. DOI: https://doi.org/10.1007/978-3-540-32243-6_11.

[34] B. Edmonds, 'Simplicity is not truth-indicative', *Philosophy and Complexity. World Scientific*, pp. 65–80, Feb. 2007. DOI: https://doi.org/10.1142/9789812707420_0005.

[35] Z. Sun, I. Lorscheid, J. Millington *et al.*, 'Simple or complicated agent-based models? a complicated issue', *Environmental Modelling & Software*, vol. 86, pp. 56–67, Dec. 2016, ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2016.09.006.

[36] T. Naylor and J. Finger, 'Verification of computer simulation models', *Management science*, vol. 14, no. 2, B–92, Oct. 1967. DOI: 10.1287/mnsc.14.2.b92.

[37] J. Kleijnen, 'Verification and validation of simulation models', *European Journal of Operational Research*, vol. 82, no. 1, pp. 145–162, Apr. 1995. DOI: 10.1016/0377-2217(94)00016-6.

[38] B. Zeigler, A. Muzy and E. Kofman, *Theory of Modeling and Simulation: Discrete Event & Iterative System Computational Foundations*, 3rd. USA: Academic Press, Inc., 2018, ISBN: 0128133708.

[39] C. Beisbart, 'Should validation and verification be separated strictly?', in *Simulation Foundations, Methods and Applications*, Springer, 2019, pp. 1005–1028. DOI: 10.1007/978-3-319-70766-2_42.

[40] A. Law, 'How to build valid and credible simulation models', in *2019 Winter Simulation Conference (WSC)*, IEEE, Dec. 2019, pp. 1402–1414. DOI: 10.1109/wsc40007.2019.9004789.

[41] O. Morgenstern and J. Von Neumann, 'Theory of games and economic behaviour.', *Economica*, vol. 13, no. 50, p. 136, May 1946. DOI: 10.2307/2550081.

[42] A. Mas-Colell, P. Mas-Colell, W. D *et al.*, *Microeconomic Theory* (Oxford student edition). Oxford University Press, 1995, ISBN: 9780195073409.

[43] D. Fudenberg and J. Tirole, 'Game theory.', *Economica*, vol. 60, no. 238, p. 245, May 1991. DOI: 10.2307/2554596.

[44] F. Hayek, 'Die verfassung der freiheit [the constitution of liberty]', *Tubingen: Mohr*, 1971.

[45] L. Hurwicz, 'INSTITUTIONS AS FAMILIES OF GAME FORMS', *The Japanese Economic Review*, vol. 47, no. 2, pp. 113–132, Jun. 1996. DOI: 10.1111/j.1468-5876.1996.tb00038.x.

[46] J. Doyle and R. Thomason, 'Background to qualitative decision theory', *AI magazine*, vol. 20, no. 2, pp. 55–55, 1999.

[47] W. Güth, R. Schmittberger and B. Schwarze, 'An experimental analysis of ultimatum bargaining', *Journal of Economic Behavior & Organization*, vol. 3, no. 4, pp. 367–388, Dec. 1982. DOI: 10.1016/0167-2681(82)90011-7.

[48] H. Oosterbeek, R. Sloof and G. Van De Kuilen, 'Cultural differences in ultimatum game experiments: Evidence from a meta-analysis', *Experimental economics*, vol. 7, no. 2, pp. 171–188, 2004.

[49] H. Simon, 'Theories of bounded rationality', *Decision and organization*, vol. 1, no. 1, pp. 161–176, 1972.

[50] J. Smith, *Evolution and the Theory of Games*. Cambridge University Press, Oct. 1982. DOI: 10.1017/cbo9780511806292.

[51] N. Bulling, 'A survey of multi-agent decision making', *KI-Künstliche Intelligenz*, vol. 28, no. 3, pp. 147–158, Jul. 2014. DOI: 10.1007/s13218-014-0314-3.

[52] L. Savage, 'The foundations of statistics', *The Mathematical Gazette*, vol. 57, no. 401, p. 220, Oct. 1973. DOI: 10.2307/3615588.

[53] V. Mazalov and J. Chirkova, *Networking Games: Network Forming Games and Games on Networks*. Academic Press, 2019.

[54] X. Song, W. Jiang, X. Liu, H. Lu, Z. Tian and X. Du, 'A survey of game theory as applied to social networks', *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 734–742, Dec. 2020, ISSN: 1007-0214. DOI: `10.26599/tst.2020.9010005`.

[55] R. Rosenthal, 'A class of games possessing pure-strategy nash equilibria', *International Journal of Game Theory*, vol. 2, no. 1, pp. 65–67, Dec. 1973. DOI: `10.1007/bf01737559`.

[56] D. Braess, 'Über ein paradoxon aus der verkehrsplanung', *Unternehmensforschung Operations Research - Recherche Opérationnelle*, vol. 12, no. 1, pp. 258–268, Dec. 1968. DOI: `10.1007/bf01918335`.

[57] P. Samuelson, 'Tragedy of the commons: Efficiency rents to the rescue of free-road inefficiencies and paradoxes', in *Does Economic Space Matter?*, Palgrave Macmillan, 1993, pp. 71–80. DOI: `10.1007/978-1-349-22906-2_4`.

[58] E. Koutsoupias, M. Mavronicolas and P. Spirakis, 'Approximate equilibria and ball fusion', *Theory of Computing Systems*, vol. 36, no. 6, pp. 683–693, Oct. 2003. DOI: `10.1007/s00224-003-1131-5`.

[59] J. Nash *et al.*, 'Equilibrium points in n-person games', in *Classics in Game Theory*, vol. 36, Princeton University Press, Nov. 1950, pp. 3–4. DOI: `10.2307/j.ctv173f1fh.6`.

[60] N. Nisan, T. Roughgarden, E. Tardos and V. Vazirani, 'Algorithmic game theory', *Communications of the ACM*, vol. 53, no. 7, pp. 78–86, Jul. 2010. DOI: `10.1145/1785414.1785439`.

[61] E. Koutsoupias and C. Papadimitriou, 'Worst-case equilibria', in *STACS 99*, Springer, 1999, pp. 404–413. DOI: `10.1007/3-540-49116-3_38`.

[62] E. Ostrom, 'An agenda for the study of institutions', *Public choice*, vol. 48, no. 1, pp. 3–25, 1986, ISSN: 1573-7101. DOI: `10.1007/bf00239556`.

[63] M. Èihák *et al.*, 'The 2007 nobel prize in economics: Mechanism design theory', *Czech Journal of Economics and Finance (Finance a uver)*, vol. 58, no. 01-02, pp. 82–89, 2008.

[64] L. Hurwicz, 'Optimality and informational efficiency in resource allocation processes', in *Studies in Resource Allocation Processes*, Cambridge University Press, Nov. 1977, pp. 393–460. DOI: 10.1017/cbo9780511752940.014.

[65] L. Hurwicz and S. Reiter, *Designing economic mechanisms*. Cambridge University Press, 2006.

[66] R. Myerson, 'Perspectives on mechanism design in economic theory', *American Economic Review*, vol. 98, no. 3, pp. 586–603, 2008.

[67] Y. Narahari, *Game theory and mechanism design*. World Scientific, 2014, vol. 4.

[68] D. Luc, 'Pareto optimality', *Pareto optimality, game theory and equilibria*, pp. 481–515, 2008.

[69] R. Sargent, 'Verification and validation of simulation models', in *Proceedings of the 2010 Winter Simulation Conference*, IEEE, Dec. 2010, pp. 166–183. DOI: 10.1109/wsc.2010.5679166.

[70] O. Balci, 'Verification validation and accreditation of simulation models', in *Proceedings of the 29th conference on Winter simulation*, ACM Press, 1997, pp. 135–141. DOI: 10.1145/268437.268462.

[71] B. Roungas, S. Meijer and A. Verbraeck, 'A framework for optimizing simulation model validation & verification', *International Journal on Advances in Systems and Measurements*, vol. 11, no. 1-2, pp. 137–152, 2018.

[72] S. Robinson, 'Simulation model verification and validation: Increasing the users' confidence', in *Proceedings of the 29th conference on Winter simulation*, ACM Press, 1997, pp. 53–59. DOI: 10.1145/268437.268448.

[73] O. Balci and R. Sargent, 'Some examples of simulation model validation using hypothesis testing', Institute of Electrical and Electronics Engineers (IEEE), Tech. Rep., 1982.

[74] O. Balci and R. Sargent, 'Validation of multivariate response models using hotelling's two-sample t2 test', *Simulation*, vol. 39, no. 6, pp. 185–192, Dec. 1982. DOI: 10.1177/003754978203900602.

[75] O. Balci and R. Sargent, 'Validation of simulation models via simultaneous confidence intervals', *American Journal of Mathematical and Management Sciences*, vol. 4, no. 3-4, pp. 375–406, Feb. 1984. DOI: 10.1080/01966324.1984.10737151.

[76] L. Passos, R. Rossetti and Z. Kokkinogenis, 'Towards the next-generation traffic simulation tools: A first appraisal', in *6th Iberian Conference on Information Systems and Technologies (CISTI 2011)*, IEEE, 2011, pp. 1–6.

[77] P. Lopez, M. Behrisch, L. Bieker-Walz *et al.*, 'Microscopic traffic simulation using SUMO', in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 2575–2582. DOI: 10.1109/itsc.2018.8569938.

[78] B. Edmonds, C. Le Page, M. Bithell *et al.*, 'Different modelling purposes', *Journal of Artificial Societies and Social Simulation*, vol. 22, no. 3, p. 6, 2019, ISSN: 1460-7425. DOI: 10.18564/jasss.3993.

[79] M. Haklay and P. Weber, 'OpenStreetMap: User-generated street maps', *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct. 2008. DOI: 10.1109/mprv.2008.80.

[80] H. Zheng, Y. Son, Y. Chiu *et al.*, 'A primer for agent-based simulation and modeling in transportation applications', United States. Federal Highway Administration, Tech. Rep., 2013.

[81] J. Müller, *The design of intelligent agents: a layered approach*. Springer Science & Business Media, 1996, vol. 1177.

[82] Y. Shoham, 'Agent-oriented programming', *Artificial intelligence*, vol. 60, no. 1, pp. 51–92, Mar. 1993. DOI: 10.1016/0004-3702(93)90034-9.

[83] M. Wooldridge, 'Agent-based software engineering', *IEE Proceedings - Software Engineering*, vol. 144, no. 1, p. 26, 1997. DOI: 10.1049/ip-sen:19971026.

[84] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Prentice Hall Upper Saddle River, NJ, USA: 2002.

[85] A. Dorri, S. Kanhere and R. Jurdak, 'Multi-agent systems: A survey', *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018. DOI: 10.1109/access.2018.2831228.

[86] J. Ferber and G. Weiss, *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley Reading, 1999, vol. 1.

[87] Y. Shoham and K. Leyton-Brown, *Multiagent Systems*. Cambridge University Press, 2008. DOI: 10.1017/cbo9780511811654.

[88] K. Clarke, 'Cellular automata and agent-based models', in *Handbook of Regional Science*, Springer, 2018, pp. 1–16. DOI: 10.1007/978-3-642-36203-3_63-1.

[89] T. Balke and N. Gilbert, 'How do agents make decisions? a survey', *Journal of Artificial Societies and Social Simulation*, vol. 17, no. 4, p. 13, 2014. DOI: 10.18564/jasss.2687.

[90] Y. Rizk, M. Awad and E. Tunstel, 'Decision making in multiagent systems: A survey', *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 514–529, Sep. 2018. DOI: 10.1109/tcds.2018.2840971.

[91] T. Nielsen and F. Jensen, *Bayesian networks and decision graphs*. Springer, 2009.

[92] R. Bellman, 'A markovian decision process', *Indiana University Mathematics Journal*, vol. 6, no. 4, pp. 679–684, 1957. DOI: 10.1512/iumj.1957.6.56038.

[93] S. Tan and J. Pearl, 'Qualitative decision theory', in *AAAI*, vol. 928, 1994.

[94] C. Boutilier, 'Toward a logic for qualitative decision theory', in *Principles of Knowledge Representation and Reasoning*, Elsevier, 1994, pp. 75–86. DOI: 10.1016/b978-1-4832-1452-8.50104-4.

[95] D. Dubois, H. Fargier and P. Perny, 'Qualitative decision theory with preference relations and comparative uncertainty: An axiomatic approach', *Artificial Intelligence*, vol. 148, no. 1-2, pp. 219–260, 2003. DOI: 10.1016/s0004-3702(03)00037-7.

[96] M. Wellman, 'Formulation of tradeoffs in planning under uncertainty', Ph.D. dissertation, Massachusetts Institute of Technology, 1988.

[97] J. McCarthy, 'Ascribing mental qualities to machines.', Stanford University, Department of Computer Science, Tech. Rep., 1979. [Online]. Available: http://jmc.stanford.edu/articles/ascribing/ascribing.pdf.

[98] A. Newell, 'The knowledge level: Presidential address', *AI magazine*, vol. 2, no. 2, pp. 1–1, 1981.

[99] R. Brafman and M. Tennenholtz, 'Modeling agents as qualitative decision makers', *Artificial Intelligence*, vol. 94, no. 1-2, pp. 217–268, 1997. DOI: 10.1016/s0004-3702(97)00024-6.

[100] H. Levesque, 'Making believers out of computers', *Artificial Intelligence*, vol. 30, no. 1, pp. 81–108, 1986. DOI: 10.1016/0004-3702(86)90068-8.

[101] M. Bratman, D. Israel and M. Pollack, 'Plans and resource-bounded practical reasoning', *Computational Intelligence*, vol. 4, no. 3, pp. 349–355, Sep. 1988. DOI: 10.1111/j.1467-8640.1988.tb00284.x.

[102] M. Georgeff, B. Pell, M. Pollack, M. Tambe and M. Wooldridge, 'The belief-desire-intention model of agency', in *Intelligent Agents V: Agents Theories, Architectures, and Languages*, Springer, 1999, pp. 1–10. DOI: 10.1007/3-540-49057-4_1.

[103] S. Akter, M. Mamun, J. Mwakalonge, G. Comert and S. Siuhi, 'A policy review of electric personal assistive mobility devices', *Transportation Research Interdisciplinary Perspectives*, vol. 11, 2021, ISSN: 2590-1982. DOI: https://doi.org/10.1016/j.trip.2021.100426. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590198221001329.

[104] K. Rand. and C. Fleming, 'An interdisciplinary review to develop guidelines for modeling population displacement as a function of infrastructure reconstruction decisions', *Transportation Research Interdisciplinary Perspectives*, vol. 3, 2019, ISSN: 2590-1982. DOI: https://doi.org/10.1016/j.trip.2019.100072.

[105] U. Wilensky *et al.*, 'Netlogo', in *NetLogo (and NetLogo User Manual)*, Center for connected learning and computer-based modeling, Northwestern University, 1999.

[106] A. Horni, K. Nagel and K. Axhausen, *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, 2016. DOI: 10.5334/baw.

[107] A. Bazzan, M. do Amarante, T. Sommer and A. Benavides, 'ITSUMO: An agent-based simulator for ITS applications', in *Proc. of the 4th Workshop on Artificial Transportation Systems and Simulation. IEEE*, 2010, p. 8.

[108] M. Treiber and A. Kesting, 'An open-source microscopic traffic simulator', *IEEE Intelligent Transportation Systems Magazine, Vol. 2(3), 6-13 (2010)*, vol. 2, no. 3, pp. 6–13, 22nd Dec. 2010. DOI: 10.1109/MITS.2010.939208.

[109] M. Guériau, R. Billot, N. El Faouzi, J. Monteil, F. Armetta and S. Hassas, 'How to assess the benefits of connected vehicles? a simulation framework for the design of cooperative traffic management strategies', *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 266–279, 2016. DOI: 10.1016/j.trc.2016.01.020.

[110] B. Torabi, M. Al-Zinati and R. Wenkstern, 'MATISSE 3.0: A large-scale multi-agent simulation system for intelligent transportation systems', in *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, Springer, 2018, pp. 357–360. DOI: 10.1007/978-3-319-94580-4_38.

[111] J. Auld, M. Hope, H. Ley, V. Sokolov, B. Xu and K. Zhang, 'POLARIS: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations', *Transportation Research Part C: Emerging Technologies*, vol. 64, pp. 101–116, 2016. DOI: 10.1016/j.trc.2015.07.017.

[112] M. Jakob and Z. Moler, 'Modular framework for simulation modelling of interaction-rich transport systems', in *16th International IEEE Conference on Intelligent*

Transportation Systems (ITSC 2013), IEEE, 2013, pp. 2152–2159. DOI: 10.1109/itsc.2013.6728547.

[113] P. Waddell, A. Borning, H. Ševčıéková and D. Socha, 'Opus (the open platform for urban simulation) and urbansim 4', in *Proceedings of the 2006 International Conference on Digital Government Research*, Digital Government Society of North America, 2006, pp. 360–361. DOI: 10.1145/1146598.1146702.

[114] G. Czura, P. Taillandier, P. Tranouez and É. Daudé, 'Mosaiic: City-level agent-based traffic simulation adapted to emergency situations', in *Proceedings of the International Conference on Social Modeling and Simulation, plus Econophysics Colloquium 2014*, Springer, 2015, pp. 265–274. DOI: 10.1007/978-3-319-20591-5_24.

[115] J. Weyl, D. Glake and T. Clemen, 'Agent-based traffic simulation at city scale with mars', in *Proceedings of the Agent-Directed Simulation Symposium*, Society for Computer Simulation International, 2018, ISBN: 9781510860131.

[116] M. Adnan, F. Pereira, C. Azevedo *et al.*, 'Simmobility: A multi-scale integrated agent-based simulation platform', in *95th Annual Meeting of the Transportation Research Board Forthcoming in Transportation Research Record*, 2016.

[117] P. Hidas, 'SITRAS: A simulation model for ITS applications', in *Towards the new together. Proceedings of the 5th World Congress on Intelligent Transport Systems*, ITS America, 1998.

[118] A. Champion, S. Éspié and J. Auberlet, 'Behavioral road traffic simulation with archisim', in *Summer Computer Simulation Conference*, Society for Computer Simulation International, 2001, pp. 359–364.

[119] Y. Xu, H. Aydt and M. Lees, 'SEMSim: A distributed architecture for multi-scale traffic simulation', in *2012 ACM/IEEE/SCS 26th Workshop on Principles of Advanced and Distributed Simulation*, IEEE, 2012, pp. 178–180. DOI: 10.1109/pads.2012.40.

[120] C. Tao and S. Huang, 'An extensible multi-agent based traffic simulation system', in *2009 International Conference on Measuring Technology and Mechatronics Automation*, vol. 3, IEEE, 2009, pp. 713–716. DOI: `10.1109/icmtma.2009.42`.

[121] T. Osogami, T. Imamichi, H. Mizuta *et al.*, 'IBM mega traffic simulator', IBM, Tech. Rep. RT0896, 2012. [Online]. Available: `https://dominoweb.draco.res.ibm.com/reports/paper.pdf`.

[122] A. D. Dumbuya, R. Wood, T. Gordon and P. Thomas, *An agent-based traffic simulation framework to model intelligent virtual driver behaviour*, 2002. [Online]. Available: `https://hdl.handle.net/2134/3161`.

[123] M. Zargayouna, B. Zeddini, G. Scemama and A. Othman, 'Simulating the impact of future internet on multimodal mobility', in *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, 2014, pp. 230–237. DOI: `10.1109/aiccsa.2014.7073203`.

[124] G. Chaurasia, B. Selvamani, N. Gupta and S. Kumar, 'Virtual chaotic traffic simulation', in *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, ACM Press, 2010, pp. 337–344. DOI: `10.1145/1924559.1924604`.

[125] G. Nakamiti, V. da Silva, J. Ventura and J. Gonçalves, 'An agent-based simulation system for traffic control in the brazilian intelligent cities project context', in *Proceedings of the 2012 Symposium on Agent Directed Simulation*, Society for Computer Simulation International, 2012, ISBN: 9781618397836. DOI: `https://dl.acm.org/doi/10.5555/2338776.2338779`.

[126] D. Pathania, B. Vissapragada, N. Jain, A. Khare, S. Lanka and K. Karlapalem, 'MUST: Multi agent simulation of multi-modal urban traffic', in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1397–1398, ISBN: 9781450319935. DOI: `https://dl.acm.org/doi/10.5555/2484920.2485243`.

[127] J. Levesque, F. Cazzolato, J. Perron, J. Hogan, T. Garneau and B. Moulin, 'CAMiCS: Civilian activity modelling in constructive simulation', in *Proceedings of the 2008 Spring Simulation Multiconference*, Society for Computer Simulation International, 2008, pp. 739–744, ISBN: 1565553195. [Online]. Available: https://dl.acm.org/doi/10.5555/1400549.1400667.

[128] J. Clymer, 'Simulation of a vehicle traffic control network using a fuzzy classifier system', in *Proceedings 35th Annual Simulation Symposium. SS 2002*, IEEE, 2002, pp. 285–291.

[129] N. Schurr, J. Marecki, J. Lewis, M. Tambe and P. Scerri, 'The defacto system: Coordinating human-agent teams for the future of disaster response', in *Multi-Agent Programming: Languages, Platforms and Applications.* Springer, 2005, pp. 197–215, ISBN: 978-0-387-26350-2. DOI: 10.1007/0-387-26350-0_8. [Online]. Available: https://doi.org/10.1007/0-387-26350-0_8.

[130] A. Banos and A. Charpentier, 'Simulating pedestrian behavior in subway stations with agents', in *Proceedings of the 4th European Social Simulation Association*, Citeseer, 2007, pp. 611–621.

[131] A. Ion, V. Constantinescu and M. Patrascu, 'An agent based simulation model applied to emergency vehicles in high traffic urban environments', in *2015 Annual Global Online Conference on Information and Computer Technology (GOCICT)*, IEEE, 2015, pp. 104–108. DOI: 10.1109/gocict.2015.26.

[132] D. Handford, A. Rogers and K. Cross, 'Agent-based traffic operator training environments for evacuation scenarios', in *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 2, IEEE, 2011, pp. 438–441. DOI: 10.1109/wi-iat.2011.202.

[133] L. Zhou and K. Zhao, 'The design of agent-based intelligent traffic visualized simulation system', in *2010 International Conference on Electrical and Control Engineering*, IEEE, 2010, pp. 3066–3069. DOI: 10.1109/icece.2010.747.

[134] F. Klügl, R. Herrler and M. Fehler, 'SeSAm: Implementation of agent-based simulation using visual programming', in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, ACM Press, 2006, pp. 1439–1440. DOI: 10.1145/1160633.1160904.

[135] J. Yoo, H. Jeong, B. Yoo, S. Kim and C. Park, 'IMAGES: Intelligent multi-agent system for freeway traffic flow simulation', in *2009 International Conference on Information Networking*, IEEE, 2009, pp. 1–5.

[136] C. Chan, B. Wang, J. Bachan and J. Macfarlane, 'Mobiliti: Scalable transportation simulation using high-performance parallel computing', in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 634–641. DOI: 10.1109/itsc.2018.8569397.

[137] T. Wang, S. Tang and P. Pang, '3d urban traffic system simulation based on geo-data', in *ITRE 2004. 2nd International Conference Information Technology: Research and Education*, IEEE, 2004, pp. 59–63. DOI: 10.1109/ITRE.2004.1393646.

[138] F. Chen and H. Pang, 'Study of multi-agent area coordination control for urban traffic', in *2008 7th World Congress on Intelligent Control and Automation*, IEEE, 2008, pp. 4046–4050. DOI: 10.1109/wcica.2008.4594510.

[139] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez and V. Koltun, 'CARLA: An open urban driving simulator', in *Proceedings of the 1st Annual Conference on Robot Learning*, vol. 78, PMLR, 2017, pp. 1–16. [Online]. Available: https://proceedings.mlr.press/v78/dosovitskiy17a.html.

[140] M. Radecký and P. Gajdoš, 'Intelligent agents for traffic simulation', in *Proceedings of the 2008 Spring Simulation Multiconference*, Society for Computer Simulation International, 2008, pp. 109–115, ISBN: 1565553195.

[141] P. Salvini and E. Miller, 'ILUTE: An operational prototype of a comprehensive microsimulation model of urban systems', *Networks and Spatial Economics*, vol. 5, no. 2, pp. 217–234, 2005. DOI: 10.1007/s11067-005-2630-5.

[142] M. Batty, C. Vargas, D. Smith, J. Serras, J. Reades and A. Johansson, 'SIMULACRA: Fast land-use—transportation models for the rapid assessment of urban futures', *Environment and Planning B: Planning and Design*, vol. 40, no. 6, pp. 987–1002, 2013. DOI: 10.1068/b4006mb.

[143] F. Wang, 'Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications', *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630–638, 2010. DOI: 10.1109/tits.2010.2060218.

[144] P. Cai, Y. Lee, Y. Luo and D. Hsu, 'SUMMIT: A simulator for urban driving in massive mixed traffic', in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2020, pp. 4023–4029. DOI: 10.1109/ICRA40945.2020.9197228.

[145] V. Chu, J. Görmer and J. Müller, 'ATSim: Combining aimsum and jade for agent-based traffic simulation', in *Proceedings of the 14th Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*, Springer, 2011.

[146] S. Thulasidasan, S. Kasiviswanathan, S. Eidenbenz, E. Galli, S. Mniszewski and P. Romero, 'Designing systems for large-scale, discrete-event simulations: Experiences with the fasttrans parallel microsimulator', in *2009 International Conference on High Performance Computing (HiPC)*, IEEE, 2009, pp. 428–437. DOI: 10.1109/hipc.2009.5433183.

[147] F. Bellifemine, F. Bergenti, G. Caire and A. Poggi, 'Jade — a java agent development framework', in *Multi-Agent Programming*, Springer, 2005, pp. 125–147. DOI: 10.1007/0-387-26350-0_5.

[148] Texas Transportation Institute, 'Early deployment of transims: Issue paper', Federal Highway Administration Washington, DC, Tech. Rep., 1999.

[149] D. Krajzewicz, G. Hertkorn, C. Rössel and P. Wagner, 'SUMO (simulation of urban mobility)-an open-source traffic simulation', in *Proceedings of the 4th middle*

East Symposium on Simulation and Modelling (MESM20002), SCS European Publishing House, 2002, pp. 183–187.

[150] M. Miska, E. Santos, E. Chung and H. Prendinger, 'Opentraffic-an open source platform for traffic simulation', in *Australasian Transport Research Forum 2011 Proceedings*, The Planning and Transport Research Centre (PATREC), 2011.

[151] G. Tamminga, P. K. and J. Van Lint, 'Open traffic: A toolbox for traffic research', *Procedia Computer Science*, vol. 32, pp. 788–795, 2014. DOI: 10.1016/j.procs.2014.05.492.

[152] E. Cornelis and L. Platbrood, 'PACSIM: A dynamic, behavioural and multimodal urban traffic simulation model', in *AIS'2002*, SCS, The society for Modeling and Simulation International, 2002, pp. 81–85.

[153] M. Fellendorf, 'VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority', in *64th Institute of Transportation Engineers Annual Meeting*, Springer, vol. 32, 1994, pp. 1–9.

[154] J. Barceló and J. Casas, 'Dynamic network simulation with AIMSUN', in *Simulation Approaches in Transportation Analysis: Recent Advances and Challenges.* Springer, 2005, pp. 57–98, ISBN: 978-0-387-24109-8. DOI: 10.1007/0-387-24109-4_3. [Online]. Available: https://doi.org/10.1007/0-387-24109-4_3.

[155] G. Cameron and G. Duncan, 'PARAMICS - parallel microscopic simulation of road traffic', *The Journal of Supercomputing*, vol. 10, no. 1, pp. 25–53, 1996. DOI: 10.1007/bf00128098.

[156] Q. Yang, H. Koutsopoulos and M. Ben-Akiva, 'Simulation laboratory for evaluating dynamic traffic management systems', *Transportation Research Record*, vol. 1710, no. 1, pp. 122–130, 2000. DOI: 10.3141/1710-14.

[157] J. Miller and E. Horowitz, 'FreeSim - a free real-time freeway traffic simulator', in *2007 IEEE Intelligent Transportation Systems Conference*, IEEE, 2007, pp. 18–23. DOI: 10.1109/itsc.2007.4357627.

[158] L. Owen, Y. Zhang, L. Rao and G. McHale, 'Traffic flow simulation using CORSIM', in *2000 Winter Simulation Conference Proceedings*, IEEE, vol. 2, 2000, pp. 1143–1147. DOI: 10.1109/WSC.2000.899077.

[159] J. Lei, K. Redmill and U. Ozguner, 'VATSIM: A simulator for vehicles and traffic', in *2001 IEEE Intelligent Transportation Systems Proceedings*, IEEE, 2001, pp. 686–691.

[160] R. Liu, 'Traffic simulation with DRACULA', in *Fundamentals of Traffic Simulation*, Springer, 2010, pp. 295–322. DOI: 10.1007/978-1-4419-6142-6_8.

[161] Y. Wang, M. Papageorgiou and A. Messmer, 'RENAISSANCE – a unified macroscopic model-based approach to real-time freeway network traffic surveillance', *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 3, pp. 190–212, 2006. DOI: 10.1016/j.trc.2006.06.001.

[162] D. Sorenson and J. Collins, 'Practical applications of traffic simulation using simtraffic', in *Compendium of Papers. Institute of Transportation Engineers 2000, District 6 Annual Meeting*, Institute of Transportation Engineers, 2000.

[163] M. Ben-Akiva, M. Bierlaire, H. Koutsopoulos and R. Mishalani, 'DynaMIT: A simulation-based system for traffic prediction', in *DACCORD short term forecasting workshop*, 1998, pp. 1–12.

[164] H. Mahmassani and S. Peeta, 'Network performance under system optimal and user equilibrium dynamic assignments: Implications for advanced traveler information systems', *Transportation Research Record*, vol. 1408, p. 83, 1993.

[165] Q. Yang and H. Koutsopoulos, 'A microscopic traffic simulator for evaluation of dynamic traffic management systems', *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 3, pp. 113–129, 1996. DOI: 10.1016/s0968-090x(96)00006-x.

[166] Bentley Systems, *CUBE voyager: Predictive modeling and simulation of transportation*, 2021. [Online]. Available: https://www.bentley.com/en/products/product-line/mobility-simulation-and-analytics/cube-voyager.

[167] H. Wallentowitz, D. Neunzig and J. Ludmann, 'Effects of new vehicle and traffic technologies — analysis of traffic flow, fuel consumption and emissions with PELOPS', in *Traffic and Mobility*, Springer, 1999, pp. 181–191. DOI: 10.1007/978-3-642-60236-8_12.

[168] R. Balakrishna, D. Morgan, H. Slavin and Q. Yang, 'Large-scale traffic simulation tools for planning and operations management', *IFAC Proceedings Volumes*, vol. 42, no. 15, pp. 117–122, 2009. DOI: 10.3182/20090902-3-us-2007.0073.

[169] M. Mahut and M. Florian, 'Traffic simulation with dynameq', in *Fundamentals of Traffic Simulation*, Springer, 2010, pp. 323–361. DOI: 10.1007/978-1-4419-6142-6_9.

[170] H. Lieu, A. Santiago and A. Kanaan, 'Corflo. an integrated traffic simulation system for corridors', in *Traffic Management. Proceedings of the Engineering Foundation Conference*, Engineering Foundation, 1992.

[171] W. Bernhard and P. Portmann, 'Traffic simulation of roundabouts in switzerland', in *2000 Winter Simulation Conference Proceedings*, IEEE, vol. 2, 2000, pp. 1148–1153.

[172] A. Elci and A. Zambakoğlu, 'City traffic simulation package and its utilization', *ACM SIGSIM Simulation Digest*, vol. 13, no. 1-4, pp. 7–11, 1982. DOI: 10.1145/1102537.1102538.

[173] F. Martinez, J. Cano, C. Calafate and P. Manzoni, 'CityMob: A mobility model pattern generator for VANETs', in *ICC Workshops - 2008 IEEE International Conference on Communications Workshops*, IEEE, 2008, pp. 370–374. DOI: 10.1109/iccw.2008.76.

[174] J. Härri, F. Filali, C. Bonnet and M. Fiore, 'Vanetmobisim: Generating realistic mobility patterns for VANETs', in *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, ACM Press, 2006, pp. 96–97. DOI: 10.1145/1161064.1161084.

[175] O. Schulzyk, J. Bongartz, T. Bildhauer *et al.*, 'A bicycle simulator based on a motion platform in a virtual reality environment—FIVIS project', in *Advances in Medical Engineering*, Springer, 2007, pp. 323–328.

[176] P. Wang and R. Glassco, 'Enhanced THOREAU traffic simulation for intelligent transportation systems (ITS)', in *Proceedings of the 27th conference on Winter simulation*, ACM Press, 1995, pp. 1110–1115. DOI: 10.1145/224401.224781.

[177] B. Wang, M. Mahmoud, J. Cuesta, H. Close, Q. Stafford-Fraser and P. Robinson, 'Enhanced traffic simulation for improved realism in driving simulators', in *Adjunct Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ACM, 2018, pp. 170–174. DOI: 10.1145/3239092.3265962.

[178] J. Henriksen, 'SLX: The x is for extensibility [simulation software]', in *2000 Winter Simulation Conference Proceedings*, IEEE, vol. 1, 2000, pp. 183–190.

[179] H. Song and O. Min, 'Statistical traffic generation methods for urban traffic simulation', in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, IEEE, 2018, pp. 247–250. DOI: 10.23919/icact.2018.8323712.

[180] J. Creagh, 'SIM-ENG: A traffic simulation engine', in *Proceedings 32nd Annual Simulation Symposium*, IEEE, 1999, pp. 4–10. DOI: 10.1109/SIMSYM.1999.766447.

[181] D. Kwon, G. Yang, C. Lee *et al.*, 'KAIST interactive bicycle simulator', in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, vol. 3, IEEE, 2001, pp. 2313–2318.

[182] B. Zhang, L. Shang and D. Chen, 'A study on the traffic intersection vehicle emission base on urban microscopic traffic simulation model', in *2009 First International Workshop on Education Technology and Computer Science*, vol. 2, IEEE, 2009, pp. 789–794. DOI: 10.1109/ETCS.2009.438.

[183] S. Chi, J. Lee and Y. Kim, 'Discrete event modeling and simulation for traffic flow analysis', in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 1, IEEE, 1995, pp. 783–788. DOI: 10.1109/ICSMC.1995.537860.

[184] E. Hardman, 'Motorway speed control strategies using SISTM', in *Eighth International Conference on Road Traffic Monitoring and Control*, IET, 1996. DOI: 10.1049/cp:19960312.

[185] M. Van Aerde, B. Hellinga, M. Baker and H. Rakha, 'Integration: An overview of traffic simulation features', *Transportation Research Records*, 1996.

[186] K. Choi, S. Kwon and M. Suh, 'Development of MATDYMO (multi-agent for traffic simulation with vehicle dynamics model) i: Development of traffic environment', *International Journal of Automotive Technology*, vol. 7, no. 1, pp. 25–34, 2006.

[187] A. Byrne, K. Courage and C. Wallace, 'Handbook of computer models for traffic operations analysis', U.S. Department of Transportation, Federal Highway Administration, Tech. Rep., 1982.

[188] N. Taylor, 'The CONTRAM dynamic traffic assignment model', *Networks and spatial economics*, vol. 3, no. 3, pp. 297–322, 2003.

[189] T. Möller, A. Padhi, D. Pinner and A. Tschiesner, 'The future of mobility is at our doorstep', *McKinsey Center for Future Mobility*, 2019. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-future-of-mobility-is-at-our-doorstep.

[190] M. Association, *Matsim - multi-agent transport simulation*, accessed on 2023-02-14, 2023. [Online]. Available: https://www.matsim.org/.

[191] C. Dobler and K. Axhausen, 'Design and implementation of a parallel queue-based traffic flow simulation', *Arbeitsberichte Verkehrs-und Raumplanung*, vol. 732, 2011. DOI: 20.500.11850/40273.

[192] D. Charypar, ‘Efficient algorithms for the microsimulation of travel behavior in very large scenarios’, Ph.D. dissertation, ETH, Eidgenössische Technische Hochschule Zürich, 2008.

[193] R. Koning, W. Tan and A. van Nes, ‘Assessing spatial configurations and transport energy usage for planning sustainable communities’, *Sustainability*, vol. 12, no. 19, p. 8146, 2020. DOI: 10.3390/su12198146.

[194] T. Novosel, L. Perković, M. Ban *et al.*, ‘Agent based modelling and energy planning – utilization of MATSim for transport energy demand modelling’, *Energy*, vol. 92, pp. 466–475, 2015. DOI: 10.1016/j.energy.2015.05.091.

[195] A. Horni and K. Axhausen, ‘MATSim agent heterogeneity and a one-week scenario’, *Arbeitsberichte Verkehrs-und Raumplanung*, vol. 836, 2012. DOI: 10.3929/ethz-b-000061926.

[196] G. Flötteröd and B. Kickhöfer, ‘Choice models in MATSim’, in *The Multi-Agent Transport Simulation MATSim*, Ubiquity Press, 2016, pp. 337–346. DOI: 10.5334/baw.49.

[197] J. Auld, M. Hope, H. Ley, B. Xu, K. Zhang and V. Sokolov, ‘Modelling framework for regional integrated simulation of transportation network and activity-based demand (polaris)’, in *Proceedings of the International Symposium for Next Generation Infrastructure*, University of Wollongong, SMART Infrastructure Facility, 2014. DOI: 10.14453/isngi2013.proc.43.

[198] T. Cokyasar, J. Auld, M. Javanmardi, O. Verbas and F. de Souza, ‘Analyzing energy and mobility impacts of privately-owned autonomous vehicles’, in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2020, pp. 1–6. DOI: 10.1109/itsc45102.2020.9294218.

[199] E. Islam, A. Moawad, J. Auld, D. Karbowski and A. Rousseau, ‘Impact of advanced vehicle technologies on energy consumption for the city of detroit using transportation system simulations’, in *2017 IEEE Vehicle Power and Propulsion Conference (VPPC)*, IEEE, 2017, pp. 1–6. DOI: 10.1109/vppc.2017.8330944.

[200] J. Auld and A. Mohammadian, 'Framework for the development of the agent-based dynamic activity planning and travel scheduling (ADAPTS) model', *Transportation Letters*, vol. 1, no. 3, pp. 245–255, 2009. DOI: 10.3328/tl.2009.01.03.245-255.

[201] Y. Lu, M. Adnan, K. Basak *et al.*, 'Simmobility mid-term simulator: A state of the art integrated agent based demand and supply model', in *94th Annual Meeting of the Transportation Research Board, Washington, DC*, 2015.

[202] R. Gopalakrishnan, A. R. Alho, T. Sakai, Y. Hara, L. Cheah and M. Ben-Akiva, 'Assessing overnight parking infrastructure policies for commercial vehicles in cities using agent-based simulation', *Sustainability*, vol. 12, p. 2673, 2020. DOI: 10.3390/su12072673.

[203] K. Marczuk, H. Hong, C. Azevedo, M. Adnan, S. Pendleton, E. Frazzoli *et al.*, 'Autonomous mobility on demand in SimMobility: Case study of the central business district in singapore', in *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, IEEE, 2015, pp. 167–172. DOI: 10.1109/iccis.2015.7274567.

[204] C. Azevedo, N. Deshmukh, B. Marimuthu *et al.*, 'SimMobility short-term: An integrated microscopic mobility simulator', *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2622, pp. 13–23, 2017. DOI: 10.3141/2622-02.

[205] G. Soares, Z. Kokkinogenis, J. Macedo and R. Rossetti, 'Agent-based traffic simulation using SUMO and JADE: An integrated platform for artificial transportation systems', in *Simulation of Urban Mobility*, Springer, 2014, pp. 44–61. DOI: 10.1007/978-3-662-45079-6_4.

[206] T. Azevedo, P. De Araújo, R. Rossetti and A. Rocha, 'JADE, TraSMAPI and SUMO: A tool-chain for simulating traffic light control', *CoRR*, vol. abs/1601.08154, 2016. arXiv: 1601.08154. [Online]. Available: http://arxiv.org/abs/1601.08154.

[207] I. Timóteo, M. Araújo, R. Rossetti and E. Oliveira, 'Using TraSMAPI for the assessment of multi-agent traffic management solutions', *Progress in Artificial Intelligence*, vol. 1, pp. 157–164, 2012. DOI: 10.1007/s13748-012-0013-y.

[208] F. Wang and S. Tang, 'A framework for artificial transportation systems: From computer simulations to computational experiments', in *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, IEEE, 2005, pp. 1130–1134.

[209] J. Wahle, A. Bazzan, F. Klügl and M. Schreckenberg, 'The impact of real-time information in a two-route scenario using agent-based simulation', *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5-6, pp. 399–417, 2002. DOI: 10.1016/s0968-090x(02)00031-1.

[210] B. Silva, A. Bazzan, G. Andriotti, F. Lopes and D. Oliveira, 'ITSUMO: An intelligent transportation system for urban mobility', in *Innovative Internet Community Systems*, Springer, 2006, pp. 224–235. DOI: 10.1007/11553762_22.

[211] R. Rossetti and R. Liu, *Advances in artificial transportation systems and simulation*. Academic Press, 2014.

[212] A. Bazzan, M. do Amarante, G. Azzi *et al.*, 'Extending traffic simulation based on cellular automata: From ParticlesTo autonomous agents', in *Proceedings of the 25th ECMS International Conference on Modelling and Simulation*, ECMS, 2011, pp. 91–97. DOI: 10.7148/2011-0091-0097.

[213] MAVS, *Agent-based intelligent traffic simulation system*, accessed on 2022-10-14, 2015. [Online]. Available: https://www.utdmavs.org/matisse/.

[214] M. Rym, G. Leask, U. Shakya and R. Steiner, 'Architectural design of the DIVAs environment', in *Proceedings of Environments for Multi-Agent Systems (E4MAS04)*, Columbia University, NY: Springer, 2004.

[215] M. Jakob, Z. Moler, A. Komenda *et al.*, 'Agentpolis: Towards a platform for fully agent-based modeling of multi-modal transportation', in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*,

International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 1501–1502. DOI: 10.5555/2343896.2344081.

[216] M. Čertický, M. Jakob, R. Pıébil and Z. Moler, 'Agent-based simulation testbed for on-demand mobility services', *Procedia Computer Science*, vol. 32, pp. 808–815, 2014. DOI: 10.1016/j.procs.2014.05.495.

[217] J. Hrnčıéř and M. Jakob, 'Generalised time-dependent graphs for fully multimodal journey planning', in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, IEEE, 2013, pp. 2138–2145. DOI: 10.1109/itsc.2013.6728545.

[218] D. Fiedler, M. Čertický, J. Alonso-Mora and M. Čáp, 'The impact of ridesharing in mobility-on-demand systems: Simulation case study in prague', in *21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 1173–1178. DOI: 10.1109/itsc.2018.8569451.

[219] A. Kesting, M. Treiber and D. Helbing, 'Agents for traffic simulation', in *Multi-Agent Systems*, vol. 11, CRC Press, 2009, pp. 325–356. DOI: 10.1201/9781420070248.ch11.

[220] M. Treiber, R. Germ and A. Kesting, 'From drivers to athletes: Modeling and simulating cross-country skiing marathons', *Traffic and Granular Flow '13*, pp. 243–249, 2015. DOI: 10.1007/978-3-319-10629-8_29.

[221] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Springer, 2013. DOI: 10.1007/978-3-642-32460-4.

[222] J. Görmer and J. Müller, 'Multiagent system architecture and method for group-oriented traffic coordination', in *6th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, IEEE, 2012, pp. 1–6. DOI: 10.1109/dest.2012.6227949.

[223] A. Dimitropoulos, W. Oueslati and C. Sintek, 'The rebound effect in road transport: A meta-analysis of empirical studies', *Energy Economics*, vol. 75, pp. 163–179, 2018, ISSN: 0140-9883. DOI: https://doi.org/10.1016/j.eneco.2018

.07.021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140988318302718.

[224] J. Jung and Y. Koo, 'Analyzing the effects of car sharing services on the reduction of greenhouse gas (GHG) emissions', *Sustainability*, vol. 10, no. 2, p. 539, 2018. DOI: 10.3390/su10020539.

[225] P. Oltermann, *Activists try to stop autobahn being built through german forest*, accessed on 2023-01-09, 2020. [Online]. Available: https://www.theguardian.com/world/2020/oct/04/activists-try-to-stop-autobahn-being-built-through-german-forest.

[226] W. Scherr, P. Manser, C. Joshi, N. Frischknecht and D. Métrailler, 'Towards agent-based travel demand simulation across all mobility choices–the role of balancing preferences and constraints', *European Journal of Transport and Infrastructure Research*, vol. 20, no. 4, pp. 152–172, 2020.

[227] J. Kamel, R. Vosooghi, J. Puchinger, F. Ksontini and G. Sirin, 'Exploring the impact of user preferences on shared autonomous vehicle modal split: A multi-agent simulation approach', *Transportation Research Procedia*, vol. 37, pp. 115–122, 2019. DOI: 10.1016/j.trpro.2018.12.173.

[228] B. Hajinasab, P. Davidsson, J. Persson and J. Holmgren, 'Towards an agent-based model of passenger transportation', in *Multi-Agent Based Simulation XVI*, Cham: Springer, 2016, pp. 132–145. DOI: 10.1007/978-3-319-31447-1_9.

[229] N. Guarino, D. Oberle and S. Staab, 'What is an ontology?', in *Handbook on Ontologies*, Berlin, Heidelberg: Springer, 2009, pp. 1–17. DOI: 10.1007/978-3-540-92673-3_0.

[230] T. Farrenkopf, M. Guckert, N. Urquhart and S. Wells, 'Ontology based business simulations', *Journal of Artificial Societies and Social Simulation*, vol. 19, no. 4, 2016. DOI: 10.18564/jasss.3266.

[231] D. McGuinness, F. Van Harmelen *et al.*, 'OWL web ontology language overview', *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.

[232] T. Berners-Lee, J. Hendler, O. Lassila *et al.*, 'The semantic web', *Scientific American*, vol. 284, no. 5, pp. 34–43, May 2001. DOI: 10.1038/scientificamerican0501-34.

[233] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean *et al.*, 'SWRL: A semantic web rule language combining OWL and RuleML', *W3C Member submission*, vol. 21, no. 79, pp. 1–31, 2004.

[234] A. Horn, 'On sentences which are true of direct unions of algebras', *Journal of Symbolic Logic*, vol. 16, no. 1, pp. 14–21, Mar. 1951. DOI: 10.2307/2268661.

[235] T. Farrenkopf, 'Applying semantic technologies to multi-agent models in the context of business simulations', Ph.D. dissertation, Edinburgh Napier University, 2017.

[236] G. Schreiber, H. Akkermans, A. Anjewierden *et al.*, *Knowledge engineering and management: the CommonKADS methodology*. Cambridge, MA: MIT Press, 2000.

[237] GfK Consumer Panels, *Consumers' choice '17 - neue Muster in der Ernährung: die Verbindung von Genuss, Gesundheit und Gemeinschaft in einer beschleunigten Welt : eine Publikation anlässlich der Anuga 2017*. Erlangen: GfK Consumer Panels and Bundesvereinigung der Deutschen Ernährungsindustrie e.V., 2017. [Online]. Available: https://www.bve-online.de/presse/infothek/publikationen-jahresbericht/consumers-choice-2017.

[238] N. Pestel and E. Sommer, 'Analyse der verteilung von einkommen und vermögen in deutschland', Institute of Labor Economics (IZA), Tech. Rep., 2016.

[239] J. Geyer, J. Nguyen, T. Farrenkopf and M. Guckert, 'AGADE traffic 2.0 - a knowledge-based approach for multi-agent traffic simulations', in *Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness. The PAAMS Collection*, Cham: Springer, 2020, pp. 417–420. DOI: 10.1007/978-3-030-49778-1_38.

[240] A. Pokahr, L. Braubach and W. Lamersdorf, 'Jadex: A BDI reasoning engine', in *Multi-Agent Programming*, Boston: Springer, 2005, pp. 149–174. DOI: 10.1007/0-387-26350-0_6.

[241] U. Engel and M. Pötschke, *Mobilität und verkehrsmittelwahl 1999/2000*, GESIS Datenarchiv, Köln. ZA4203 Datenfile Version 1.0.0, 2013. DOI: 10.4232/1.11591.

[242] Presse- und Informationsamt der Bundesregierung, Berlin, *Automobile berufspendler*, GESIS Datenarchiv, Köln. ZA6741 Datenfile Version 1.0.0, 2020. DOI: 10.4232/1.13523.

[243] M. Berger, C. Müller and N. Nüske, 'Digital nudging in online grocery stores : Towards ecologically sustainable nutrition', in *Proceedings of the 41st International Conference on Information Systems (ICIS 2020)*, Hyderabad, India: AIS Electronic Library, Dec. 2020. [Online]. Available: https://eref.uni-bayreuth.de/57729/.

[244] Statista, *Lebensmittelkauf in deutschland*, 2020. [Online]. Available: https://de.statista.com/statistik/studie/id/12521/dokument/einkauf-und-konsum-von-lebensmitteln-statista-dossier/.

[245] P. Hart, N. Nilsson and B. Raphael, 'A formal basis for the heuristic determination of minimum cost paths', *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. DOI: 10.1109/tssc.1968.300136.

[246] Statistische Ämter des Bundes und der Länder, *Zensus 2011: Methoden und Verfahren*. Wiesbaden, Hesse, Germany: Statistisches Bundesamt, 2015.

[247] N. Hirschauer, M. Lehberger and O. Musshoff, 'Happiness and utility in economic thought—or: What can we learn from happiness research for public policy analysis and public policy making?', *Social Indicators Research*, vol. 121, no. 3, pp. 647–674, 2015.

[248] D. Kahneman, P. Wakker and R. Sarin, 'Back to bentham? explorations of experienced utility', *The quarterly journal of economics*, vol. 112, no. 2, pp. 375–406, 1997.

[249] C. Pfeiffer, 'Einkaufsverhandlungen', in *Spieltheorie–Erfolgreich verhandeln im Einkauf*, Springer, 2021, pp. 79–122.

[250] L. Zheng, P. Cheng and L. Chen, 'Auction-based order dispatch and pricing in ridesharing', in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, IEEE, 2019, pp. 1034–1045.

[251] A. Kleiner, B. Nebel and V. A. Ziparo, 'A mechanism for dynamic ride sharing based on parallel auctions', in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[252] E. Kamar and E. Horvitz, 'Collaboration and shared plans in the open world: Studies of ridesharing', in *Twenty-first international joint conference on artificial intelligence*, Citeseer, 2009.

[253] P. Klemperer, *Auctions: theory and practice*. Princeton University Press, 2004.

[254] A. Barabási and R. Albert, 'Emergence of scaling in random networks', *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[255] R. Albert and A. Barabási, 'Statistical mechanics of complex networks', *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, Jan. 2002. DOI: `10.1103/revmodphys.74.47`.

[256] P. Preciado, T. Snijders, W. Burk, H. Stattin and M. Kerr, 'Does proximity matter? distance dependence of adolescent friendships', *Social Networks*, vol. 34, no. 1, pp. 18–31, 2012, ISSN: 0378-8733. DOI: `https://doi.org/10.1016/j.socnet.2011.01.002`.

[257] W. M. Gorman, 'Convex indifference curves and diminishing marginal utility', *Journal of Political Economy*, vol. 65, no. 1, pp. 40–50, Feb. 1957, ISSN: 1537-534X. DOI: `10.1086/257880`.

[258] X. Heitmann, *Anschluss per app: RMV und ioki starten europaweit einmaliges on-demand-projekt*, accessed on 2023-01-31, 2020. [Online]. Available: https://www.messe-giessen.de/veranstaltungen/konzerte-sportevents/.

[259] V. Pawlik, *Festival- und konzertbesucher*, accessed on 2023-01-18, 2022. [Online]. Available: https://de.statista.com/themen/4955/festival-und-konzert/besucher/#topicOverview.

[260] R. Schäfer, *Viele machen sich auf diesen weg*, accessed on 2023-03-14, 2022. [Online]. Available: https://www.giessener-anzeiger.de/stadt-giessen/viele-machen-sich-auf-diesen-weg-91656170.html.

[261] B. Möller, accessed on 2023-03-14, 2022. [Online]. Available: https://www.giessener-allgemeine.de/giessen/konzept-fuer-verkehrsversuch-steht-91553958.html.

[262] B. Möller, *Bauarbeiten auf hauptverbindungsstraße in gießen stehen an*, accessed on 2023-03-14, 2023. [Online]. Available: https://www.giessener-allgemeine.de/giessen/fussgaengerueberweg-kommt-im-fruehjahr-giessen-92061045.html.

[263] L. Hamill and N. Gilbert, 'Social circles: A simple structure for agent-based social network models', *Journal of Artificial Societies and Social Simulation*, vol. 12, no. 2, p. 3, 2009, ISSN: 1460-7425. [Online]. Available: https://www.jasss.org/12/2/3.html.

[264] M. Darvishi and G. Ahmadi, 'Validation techniques of agent based modelling for geospatial simulations.', *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. XL-2/W3, pp. 91–95, Oct. 2014. DOI: 10.5194/isprsarchives-xl-2-w3-91-2014.

[265] M. Remondino and G. Correndo, 'Mabs validation through repeated execution and data mining analysis', *International Journal of Simulation: Systems, Science & Technology*, vol. 7, no. 6, pp. 10–21, 2006.

[266] P. Godefroid and K. Sen, 'Combining model checking and testing', *Handbook of Model Checking*, pp. 613–649, 2018.

[267] A. Turing *et al.*, 'On computable numbers, with an application to the entscheidungsproblem', *Journal of Math*, vol. 58, no. 345-363, p. 5, 1936.

[268] E. Clarke, T. Henzinger and H. Veith, 'Introduction to model checking', *Handbook of Model Checking*, pp. 1–26, 2018.

[269] K. Popper, *The logic of scientific discovery*. Routledge, 2005, ISBN: 0–415–27843–0.

[270] E. Clarke, W. Klieber, M. Nováček and P. Zuliani, 'Model checking and the state explosion problem', *Tools for Practical Software Verification: LASER, International Summer School 2011, Elba Island, Italy, Revised Tutorial Lectures*, pp. 1–30, 2012. DOI: `10.1007/978-3-642-35746-6_1`.

[271] S. Edelkamp, V. Schuppan, D. Bošnački, A. Wijs, A. Fehnker and H. Aljazzar, 'Survey on directed model checking', in *Model Checking and Artificial Intelligence*, Springer, 2009, pp. 65–89. DOI: `10.1007/978-3-642-00431-5_5`.

[272] N. Rezaee and H. Momeni, 'A hybrid meta-heuristic approach to cope with state space explosion in model checking technique for deadlock freeness', *Journal of AI and Data Mining*, vol. 8, no. 2, pp. 189–199, 2020.

[273] P. Körner and M. Leuschel, 'Towards practical partial order reduction for high-level formalisms', in *Lecture Notes in Computer Science*, Springer, 2023, pp. 72–91. DOI: `10.1007/978-3-031-25803-9_5`.

[274] B. Cirisci, C. Enea, A. Farzan and S. Mutluergil, 'A pragmatic approach to stateful partial order reduction', in *Verification, Model Checking, and Abstract Interpretation: 24th International Conference, VMCAI 2023, Boston, MA, USA, January 16–17, 2023, Proceedings*, Springer, 2023, pp. 129–154. DOI: `10.1007/978-3-031-24950-1_7`.

[275] W. Jamroga, W. Penczek, T. Sidoruk, P. Dembiński and A. Mazurkiewicz, 'Towards partial order reductions for strategic ability', *Journal of Artificial Intelligence Research*, vol. 68, pp. 817–850, Aug. 2020. DOI: `10.1613/jair.1.11936`.

[276] Y. Phyo, C. Minh Do and K. Ogata, 'A divide & conquer approach to leads-to model checking', *The Computer Journal*, vol. 65, no. 6, pp. 1353–1364, 2022.

[277] Ł. Mikulski, W. Jamroga and D. Kurpiewski, 'Towards assume-guarantee verification of strategic ability', in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, 2022, pp. 1702–1704.

[278] M. Fisher and M. Wooldridge, 'On the formal specification and verification of multi-agent systems', *International Journal of Cooperative Information Systems*, vol. 6, no. 01, pp. 37–65, Mar. 1997. DOI: 10.1142/s0218843097000057.

[279] M. Bourahla and M. Benmohamed, 'Formal specification and verification of multi-agent systems', *Electronic Notes in Theoretical Computer Science*, vol. 123, pp. 5–17, Mar. 2005. DOI: 10.1016/j.entcs.2004.04.042.

[280] W. Jamroga and W. Penczek, 'Specification and verification of multi-agent systems', *Lectures on Logic and Computation: ESSLLI 2010 Copenhagen, Denmark, August 2010, ESSLLI 2011, Ljubljana, Slovenia, August 2011, Selected Lecture Notes*, pp. 210–263, 2012. DOI: 10.1007/978-3-642-31485-8_6.

[281] R. Bordini, L. Dennis, B. Farwer and M. Fisher, 'Automated verification of multi-agent programs', in *2008 23rd IEEE/ACM International Conference on Automated Software Engineering*, IEEE, Sep. 2008, pp. 69–78. DOI: 10.1109/ase.2008.17.

[282] M. Moreno, J. Pavón and A. Rosete, 'Testing in agent oriented methodologies', in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, Springer, 2009, pp. 138–145. DOI: 10.1007/978-3-642-02481-8_20.

[283] A. Rahman, A. Haron, S. Sahibuddin and M. Harun, 'An empirical study of the software project requirements engineering practice in malaysia public sector-a perspective from the stakeholders' challenges', *International Journal of Computer Theory and Engineering*, vol. 6, no. 1, p. 52, 2014, ISSN: 1793-8201. DOI: 10.7763/ijcte.2014.v6.836.

[284] E. Gonçalves, M. Cortés, G. Campos *et al.*, 'MAS-ML 2.0: Supporting the modelling of multi-agent systems with different agent architectures', *Journal of Systems and Software*, vol. 108, pp. 77–109, 2015.

[285] J. Odell, H. Parunak and B. Bauer, 'Extending UML for agents', in *Proceedings of the agent-oriented information systems workshop at the 17th national conference on artificial intelligence*, 2000, pp. 3–17.

[286] G. Wagner, 'The agent–object-relationship metamodel: Towards a unified view of state and behavior', *Information Systems*, vol. 28, no. 5, pp. 475–504, 2003.

[287] R. Choren and C. Lucena, 'The anote modeling language for agent-oriented specification', in *International Workshop on Software Engineering for Large-Scale Multi-agent Systems*, Springer, 2004, pp. 198–212.

[288] N. Spanoudakis and P. Moraitis, 'The agent modeling language (AMOLA)', in *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, Springer, 2008, pp. 32–44.

[289] M. Cossentino, N. Gaud, V. Hilaire, S. Galland and A. Koukam, 'ASPECS: An agent-oriented software process for engineering complex systems', *Autonomous Agents and Multi-Agent Systems*, vol. 20, no. 2, pp. 260–304, 2010.

[290] L. Padgham and M. Winikoff, 'Prometheus', in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 1 - AAMAS '02*, Springer, ACM Press, 2002, pp. 174–185. DOI: 10.1145/544741.544749.

[291] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia and J. Mylopoulos, 'Tropos: An agent-oriented software development methodology', *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, May 2004. DOI: 10.1023/b:agnt.0000018806.20944.ef.

[292] M. Wooldridge, N. Jennings and D. Kinny, 'The gaia methodology for agent-oriented analysis and design', *Autonomous Agents and multi-agent systems*, vol. 3, no. 3, pp. 285–312, 2000.

[293] L. Schruben, 'Simulation modeling with event graphs', *Communications of the ACM*, vol. 26, no. 11, pp. 957–963, 1983.

[294] A. Törn, 'Simulation graphs: A general tool for modeling simulation designs', *Simulation*, vol. 37, no. 6, pp. 187–194, 1981.

[295] B. Bekiroglu, 'A cause-effect graph software testing tool', *European Journal of Computer Science and Information Technology*, vol. 5, no. 4, pp. 11–24, 2017.

[296] F. Klügl, 'A validation methodology for agent-based simulations', in *Proceedings of the 2008 ACM symposium on Applied computing*, ACM Press, 2008, pp. 39–43. DOI: `10.1145/1363686.1363696`.

[297] A. Harper, N. Mustafee and M. Yearworth, 'Facets of trust in simulation studies', *European Journal of Operational Research*, vol. 289, no. 1, pp. 197–213, 2021, ISSN: 0377-2217. DOI: `https://doi.org/10.1016/j.ejor.2020.06.043`.

[298] L. Andras P. snd Esterle, M. Guckert, T. Han *et al.*, 'Trusting intelligent machines: Deepening trust within socio-technical systems', *IEEE Technology and Society Magazine*, vol. 37, no. 4, pp. 76–83, Dec. 2018. DOI: `10.1109/mts.2018.2876107`.

[299] G. Kiczales, J. Lamping, A. Mendhekar *et al.*, 'Aspect-oriented programming', in *ECOOP'97—Object-Oriented Programming: 11th European Conference Jyväskylä, Finland, June 9–13, 1997 Proceedings 11*, Springer, 1997, pp. 220–242. DOI: `10.1007/bfb0053381`.

[300] D. Ufuktepe, T. Ayav and F. Belli, 'Test input generation from cause–effect graphs', *Software Quality Journal*, pp. 1–50, 2021.

[301] J. Sterman, *Business dynamics.* McGraw-Hill, Inc., 2000, ISBN: 978-0071179898.

[302] F. Schoeneborn, *Linking balanced scorecard to system dynamics*, 2003.

[303] J. Pearl, 'Bayesian networks: A model of self-activated memory for evidential reasoning', in *Proceedings of the 7th conference of the Cognitive Science Society, University of California, Irvine, CA, USA*, 1985, pp. 15–17.

[304] J. Pearl, 'From bayesian networks to causal networks', in *Mathematical models for handling partial knowledge in artificial intelligence*, Springer, 1995, pp. 157–182, ISBN: 9781489914248. DOI: 10.1007/978-1-4899-1424-8_9.

[305] D. Moody, 'The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering', *IEEE Transactions on Software Engineering*, vol. 35, no. 6, pp. 756–779, Nov. 2009, ISSN: 0098-5589. DOI: 10.1109/TSE.2009.67.

[306] A. Bundy and L. Wallen, 'Breadth-first search', *Catalogue of artificial intelligence tools*, pp. 13–13, 1984. DOI: 10.1007/978-3-642-96868-6_25.

[307] N. Urquhart, S. Powers, Z. Wall, A. Fonzone, J. Ge and J. Polhill, 'Simulating the actions of commuters using a multi-agent system', *Journal of Artificial Societies and Social Simulation*, vol. 22, no. 2, p. 10, 2019, ISSN: 1460-7425. DOI: 10.18564/jasss.4007. [Online]. Available: http://jasss.soc.surrey.ac.uk/22/2/10.html.

[308] R. Dijkman, M. Dumas and C. Ouyang, 'Semantics and analysis of business process models in BPMN', *Information and Software technology*, vol. 50, no. 12, pp. 1281–1294, Nov. 2008. DOI: 10.1016/j.infsof.2008.02.006.

[309] J. O'Madadhain, D. Fisher, P. Smyth, S. White and Y. Boey, 'Analysis and visualization of network data using JUNG', *Journal of Statistical Software*, vol. 10, no. 2, pp. 1–35, 2005.

[310] D. Michail, J. Kinable, B. Naveh and J. Sichi, 'Jgrapht—a java library for graph data structures and algorithms', *ACM Transactions on Mathematical Software (TOMS)*, vol. 46, no. 2, pp. 1–29, May 2020. DOI: 10.1145/3381449.

Appendix A

List of Log Entries for the Reduced Example of the Grocery Shopping Simulation

```
threadID: 1; callNo: 0; .Entrypoint -> aspects.example.Main.main(args)
threadID: 1; callNo: 1; aspects.example.Main.main -> aspects.example.
    SimulationController1854731462.init()
threadID: 1; callNo: 2; aspects.example.Main.main -> aspects.example.
    SimulationController1854731462.performSimulation(numberOfAgents,personas)
threadID: 1; callNo: 3; aspects.example.SimulationController.performSimulation ->
    aspects.example.AgentFactory1389133897.init()
threadID: 1; callNo: 4; aspects.example.SimulationController.performSimulation ->
    aspects.example.AgentFactory1389133897.createAgents(numberOfAgents,personas)
threadID: 1; callNo: 5; aspects.example.AgentFactory.createAgents -> aspects.example
    .Agent1534030866.init(id,personaProfile)
threadID: 1; callNo: 6; aspects.example.AgentFactory.createAgents -> aspects.example
    .Agent664223387.init(id,personaProfile)
threadID: 1; callNo: 7; aspects.example.AgentFactory.createAgents -> aspects.example
    .Agent824909230.init(id,personaProfile)
threadID: 1; callNo: 8; aspects.example.AgentFactory.createAgents -> aspects.example
    .Agent122883338.init(id,personaProfile)
threadID: 1; callNo: 9; aspects.example.AgentFactory.createAgents -> aspects.example
    .Agent666641942.init(id,personaProfile)
threadID: 1; callNo: 10; aspects.example.AgentFactory.createAgents -> aspects.
    example.Agent960604060.init(id,personaProfile)
threadID: 1; callNo: 11; aspects.example.AgentFactory.createAgents -> aspects.
    example.Agent1349393271.init(id,personaProfile)
threadID: 1; callNo: 12; aspects.example.AgentFactory.createAgents -> aspects.
    example.Agent1338668845.init(id,personaProfile)
threadID: 1; callNo: 13; aspects.example.AgentFactory.createAgents -> aspects.
    example.Agent159413332.init(id,personaProfile)
threadID: 1; callNo: 14; aspects.example.AgentFactory.createAgents -> aspects.
```

```
    example.Agent1028214719.init(id,personaProfile)
threadID: 1; callNo: 15; aspects.example.SimulationController.performSimulation ->
    aspects.example.AgentFactory1389133897.getAgents()
threadID: 1; callNo: 76; aspects.example.SimulationController.performSimulation ->
    aspects.example.StatisticController1146743572.init()
threadID: 1; callNo: 77; aspects.example.SimulationController.performSimulation ->
    aspects.example.StatisticController1146743572.calculatePerformanceIndicators(
    Agent)
threadID: 1; callNo: 78; aspects.example.StatisticController.
    calculatePerformanceIndicators -> aspects.example.StatisticController1146743572.
    determineGlobalTravelDistance()
threadID: 1; callNo: 79; aspects.example.StatisticController.
    determineGlobalTravelDistance -> aspects.example.Agent1534030866.getDrivenKm()
threadID: 1; callNo: 80; aspects.example.StatisticController.
    determineGlobalTravelDistance -> aspects.example.Agent664223387.getDrivenKm()
threadID: 1; callNo: 81; aspects.example.StatisticController.
    determineGlobalTravelDistance -> aspects.example.Agent824909230.getDrivenKm()
threadID: 1; callNo: 82; aspects.example.StatisticController.
    determineGlobalTravelDistance -> aspects.example.Agent122883338.getDrivenKm()
threadID: 1; callNo: 83; aspects.example.StatisticController.
    determineGlobalTravelDistance -> aspects.example.Agent666641942.getDrivenKm()
threadID: 1; callNo: 84; aspects.example.StatisticController.
    determineGlobalTravelDistance -> aspects.example.Agent960604060.getDrivenKm()
threadID: 1; callNo: 85; aspects.example.StatisticController.
    determineGlobalTravelDistance -> aspects.example.Agent1349393271.getDrivenKm()
threadID: 1; callNo: 86; aspects.example.StatisticController.
    determineGlobalTravelDistance -> aspects.example.Agent1338668845.getDrivenKm()
threadID: 1; callNo: 87; aspects.example.StatisticController.
    determineGlobalTravelDistance -> aspects.example.Agent159413332.getDrivenKm()
threadID: 1; callNo: 88; aspects.example.StatisticController.
    determineGlobalTravelDistance -> aspects.example.Agent1028214719.getDrivenKm()
threadID: 1; callNo: 89; aspects.example.StatisticController.
    calculatePerformanceIndicators -> aspects.example.StatisticController1146743572.
    determineGlobalCombustionDistance()
threadID: 1; callNo: 90; aspects.example.StatisticController.
    determineGlobalCombustionDistance -> aspects.example.Agent1534030866.getDrivenKm
    ()
threadID: 1; callNo: 91; aspects.example.StatisticController.
    determineGlobalCombustionDistance -> aspects.example.Agent664223387.getDrivenKm
    ()
threadID: 1; callNo: 92; aspects.example.StatisticController.
    determineGlobalCombustionDistance -> aspects.example.Agent824909230.getDrivenKm
    ()
threadID: 1; callNo: 93; aspects.example.StatisticController.
    determineGlobalCombustionDistance -> aspects.example.Agent122883338.getDrivenKm
```

```
    ()
threadID: 1; callNo: 94; aspects.example.StatisticController.
    determineGlobalCombustionDistance -> aspects.example.Agent666641942.getDrivenKm
    ()
threadID: 1; callNo: 95; aspects.example.StatisticController.
    determineGlobalCombustionDistance -> aspects.example.Agent960604060.getDrivenKm
    ()
threadID: 1; callNo: 96; aspects.example.StatisticController.
    determineGlobalCombustionDistance -> aspects.example.Agent1349393271.getDrivenKm
    ()
threadID: 1; callNo: 97; aspects.example.StatisticController.
    determineGlobalCombustionDistance -> aspects.example.Agent1338668845.getDrivenKm
    ()
threadID: 1; callNo: 98; aspects.example.StatisticController.
    determineGlobalCombustionDistance -> aspects.example.Agent159413332.getDrivenKm
    ()
threadID: 1; callNo: 99; aspects.example.StatisticController.
    determineGlobalCombustionDistance -> aspects.example.Agent1028214719.getDrivenKm
    ()
threadID: 23; callNo: 18; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent1534030866.planJourney()
threadID: 23; callNo: 23; aspects.example.Agent.planJourney -> aspects.example.
    Journey1309739527.init(modes,locations)
threadID: 23; callNo: 27; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent1534030866.purchaseItems()
threadID: 23; callNo: 30; aspects.example.Agent.purchaseItems -> aspects.example.
    Agent1534030866.groceryShopping()
threadID: 23; callNo: 44; aspects.example.Agent.groceryShopping -> aspects.example.
    Agent1534030866.setDrivenKm(drivenKm)
threadID: 24; callNo: 19; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent664223387.planJourney()
threadID: 24; callNo: 22; aspects.example.Agent.planJourney -> aspects.example.
    Journey763792618.init(modes,locations)
threadID: 24; callNo: 26; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent664223387.purchaseItems()
threadID: 24; callNo: 28; aspects.example.Agent.purchaseItems -> aspects.example.
    Agent664223387.groceryShopping()
threadID: 24; callNo: 43; aspects.example.Agent.groceryShopping -> aspects.example.
    Agent664223387.setDrivenKm(drivenKm)
threadID: 25; callNo: 21; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent824909230.planJourney()
threadID: 25; callNo: 24; aspects.example.Agent.planJourney -> aspects.example.
    Journey1810876065.init(modes,locations)
threadID: 25; callNo: 31; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent824909230.purchaseItems()
```

```
threadID: 25; callNo: 32; aspects.example.Agent.purchaseItems -> aspects.example.
    Agent824909230.groceryShopping()
threadID: 25; callNo: 46; aspects.example.Agent.groceryShopping -> aspects.example.
    Agent824909230.setDrivenKm(drivenKm)
threadID: 26; callNo: 29; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent122883338.planJourney()
threadID: 26; callNo: 33; aspects.example.Agent.planJourney -> aspects.example.
    Journey1813050657.init(modes,locations)
threadID: 26; callNo: 35; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent122883338.purchaseItems()
threadID: 26; callNo: 36; aspects.example.Agent.purchaseItems -> aspects.example.
    Agent122883338.groceryShopping()
threadID: 26; callNo: 49; aspects.example.Agent.groceryShopping -> aspects.example.
    Agent122883338.setDrivenKm(drivenKm)
threadID: 27; callNo: 45; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent666641942.planJourney()
threadID: 27; callNo: 51; aspects.example.Agent.planJourney -> aspects.example.
    Journey243497748.init(modes,locations)
threadID: 27; callNo: 58; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent666641942.purchaseItems()
threadID: 27; callNo: 60; aspects.example.Agent.purchaseItems -> aspects.example.
    Agent666641942.groceryShopping()
threadID: 27; callNo: 63; aspects.example.Agent.groceryShopping -> aspects.example.
    Agent666641942.setDrivenKm(drivenKm)
threadID: 28; callNo: 52; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent960604060.planJourney()
threadID: 28; callNo: 54; aspects.example.Agent.planJourney -> aspects.example.
    Journey1736463535.init(modes,locations)
threadID: 28; callNo: 59; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent960604060.purchaseItems()
threadID: 28; callNo: 62; aspects.example.Agent.purchaseItems -> aspects.example.
    Agent960604060.groceryShopping()
threadID: 28; callNo: 64; aspects.example.Agent.groceryShopping -> aspects.example.
    Agent960604060.setDrivenKm(drivenKm)
threadID: 29; callNo: 38; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent1349393271.planJourney()
threadID: 29; callNo: 39; aspects.example.Agent.planJourney -> aspects.example.
    Journey1261706839.init(modes,locations)
threadID: 29; callNo: 47; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent1349393271.purchaseItems()
threadID: 29; callNo: 50; aspects.example.Agent.purchaseItems -> aspects.example.
    Agent1349393271.groceryShopping()
threadID: 29; callNo: 56; aspects.example.Agent.groceryShopping -> aspects.example.
    Agent1349393271.setDrivenKm(drivenKm)
threadID: 30; callNo: 40; aspects.example.SimulationController.performSimulation ->
```

```
    aspects.example.Agent1338668845.planJourney()
threadID: 30; callNo: 41; aspects.example.Agent.planJourney -> aspects.example.
    Journey204629965.init(modes,locations)
threadID: 30; callNo: 53; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent1338668845.purchaseItems()
threadID: 30; callNo: 55; aspects.example.Agent.purchaseItems -> aspects.example.
    Agent1338668845.groceryShopping()
threadID: 30; callNo: 57; aspects.example.Agent.groceryShopping -> aspects.example.
    Agent1338668845.setDrivenKm(drivenKm)
threadID: 31; callNo: 65; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent159413332.planJourney()
threadID: 31; callNo: 66; aspects.example.Agent.planJourney -> aspects.example.
    Journey935545904.init(modes,locations)
threadID: 31; callNo: 69; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent159413332.purchaseItems()
threadID: 31; callNo: 71; aspects.example.Agent.purchaseItems -> aspects.example.
    Agent159413332.groceryShopping()
threadID: 31; callNo: 72; aspects.example.Agent.groceryShopping -> aspects.example.
    Agent159413332.setDrivenKm(drivenKm)
threadID: 32; callNo: 68; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent1028214719.planJourney()
threadID: 32; callNo: 70; aspects.example.Agent.planJourney -> aspects.example.
    Journey596698459.init(modes,locations)
threadID: 32; callNo: 73; aspects.example.SimulationController.performSimulation ->
    aspects.example.Agent1028214719.purchaseItems()
threadID: 32; callNo: 74; aspects.example.Agent.purchaseItems -> aspects.example.
    Agent1028214719.groceryShopping()
threadID: 32; callNo: 75; aspects.example.Agent.groceryShopping -> aspects.example.
    Agent1028214719.setDrivenKm(drivenKm)
```