Informatics

An Agent Based Simulation of Geospatial Aspects of Street Crime in Maputo City

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Business Informatics

eingereicht von

Philipp Schnatter
Matrikelnummer 00325962

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Thomas Grechenig

Wien, 17. August 2021

Unterschrift Verfasser

Unterschrift Betreuung

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Informatics

An Agent Based Simulation of Geospatial Aspects of Street Crime in Maputo City

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Philipp Schnatter
Registration Number 00325962

to the Faculty of Informatics

at the TU Wien

Advisor: Thomas Grechenig

Vienna, 17th August, 2021

Signature Author

Signature Advisor

An Agent Based Simulation of Geospatial Aspects of Street Crime in Maputo City

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Business Informatics

eingereicht von

Philipp Schnatter
Matrikelnummer 00325962

ausgeführt am
Institut für Information Systems Engineering
Forschungsbereich Business Informatics
Forschungsgruppe Industrielle Software
der Fakultät für Informatik der Technischen Universität Wien

Betreuung: Thomas Grechenig

Wien, 17. August 2021

Erklärung zur Verfassung der Arbeit

Philipp Schnatter

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 17. August 2021

Philipp Schnatter

Kurzfassung

Die Routine-Aktivitäts-Theorie von Cohen und Felson (1979) postuliert, dass die Kriminalitätsrate sinkt, wenn Menschen mehr Zeit an sicheren Orten wie zu Hause verbringen, auch wenn die Anzahl der Täter und Opfer unverändert bleibt. Dieser Ansatz betrachtet Kriminalität als räumliche und zeitliche Konvergenz von Täter und Opfer bei gleichzeitiger Abwesenheit von Schutzpersonen. Die empirische Gültigkeit dieser Theorie ist jedoch aufgrund der notwendigen personenbezogenen Daten von Routinetätigkeiten und Kriminalität umstritten. In dieser Arbeit wird eine agentenbasierte Simulation entwickelt, um in kontrollierten Experimenten empirische Daten zu erzeugen. Diese Arbeit beschreibt die Entwicklung eines agentenbasierten Modells der Kriminalität. In der Simulation gehen Bürgeragenten der Stadt Maputo ihren täglichen Routinetätigkeiten nach und interagieren bei Straßenüberfällen. Der Einfluss von Routinetätigkeiten auf die Kriminalitätsrate und die Effektivität einer Polizeistrategie werden untersucht.

Die Routine-Aktivitäts-Theorie wird als ein valides Modell der sozialen Interaktion in Form von Straßenkriminalität bewertet. Die eingeführte Policing-Strategie reduzierte die Kriminalitätsrate um 16 %, verursachte aber eine örtliche Verschiebung der Kriminalität von 45 % der durch die Policing-Strategie zusätzlich verhinderten Delikte. Die Lösung ist ein valides und robustes Modell zur Untersuchung räumlicher Aspekte der Kriminalität und zur Erforschung von Aspekten menschlicher Interaktion auf der Grundlage einfacher agentenbasierter Regeln.

Keywords: Agentenbasierte Modellierung, Simulation, Routine-Aktivitäts-Theorie, GIS, Kriminologie, Maputo

Abstract

The routine activity theory by Cohen and Felson (1979) postulates that crime rates decrease when people spend more time at safe places like home, even when the number of offenders and victims remains unchanged. This approach regards crime as the convergence in space and time of offender, victim and the absence of a guardian. However, the empiric validity of this theory is still debated, due to the required personal data of routine activities and crime incidents. In this work, a computer simulation is developed to produce empiric data in controlled experiments. This thesis describes the development of an agent-based crime model. In the simulation, citizen agents of Maputo perform their daily routine activities and interact in street robberies. The influence of routine activities on crime rates and the effectiveness of a policing strategy are investigated.

The routine activity theory is evaluated to be valid for the presented street crime model. The introduced policing strategy reduced the crime rate by 16 %, but caused crime displacement for 45 % of additionally prevented offenses. The solution is a valid and robust model for investigating spatial aspects of crime and for exploring aspects of human interaction based on simple agent rules.

Keywords: agent-based modeling, simulation, routine, activity, GIS, criminology, Maputo

Contents

CHAPTER 1

Introduction

1.1 Problem Statement

The sociologists Lawrence E. Cohen and Marcus Felson investigated a sociological paradox: While the social and economic situation in the United States improved between 1960 and 1975, reported rates of violent and non-violent crime increased. In 1979, Cohen and Felson postulated the *routine activity theory*, explaining that the paradoxical crime rate changes relate to a shift in the routine activities of citizens away from their homes. These changes especially affect a class of crime known as street robbery. This type of offense is classified as *instrumental crime*, which is about economic gain and therefore a result of rational choice rather than an expressive crime. Cohen and Felson further elaborated that robbery requires three elements to occur: a motivated offender, a suitable target and the absence of a guardian such as police officers or civil bystanders. These three elements must converge in space and time for the offense to happen. Cohen and Felson argued that an increase in convergences leads to an increase in the number of criminal acts, even if the number of potential offenders and victims remain constant [CF79].

The Routine Activity Approach is a widely accepted theory in sociology and criminology. Crime prevention strategies based on this approach have been proven in studies and implemented successfully [Ton95]. Several studies have been conducted to evaluate the routine activity theory based on empirical data. However, due to the difficulty of obtaining personal data, the empirical validity is still debated [AS12].

Computer simulation as a scientific method is an established alternative to produce empirical data about a real phenomenon within a controlled experiment [Win03]. Simulations in science follow an experimental approach. Researchers build simulation models, run the models while varying parameters and investigate the results of different conditions. The main difference to experiments is that simulations are concerned about models of the real world, not the phenomenon itself. The goal is that conclusions about the model also apply to the target under study. The agent-based simulation (ABS) approach introduces the concept of agency to computer simulation. In social sciences, agency represents the ability of individuals to act and decide independently. The introduction of computer simulation in social sciences enables researchers to describe, investigate and understand dynamic interactions at agent level, which lead to society-level patterns (emergence).

Cohen and Felson emphasized the spatial-temporal structure of routine activities, which plays a role in determining the frequency and the location of criminal acts. As a result, the location and the geography are relevant aspects of the study. Geographic information systems (GIS) are a family of computer systems, which have the ability to work with geographic data. Crime is not spread equally across the environment. Certain neighborhoods have higher crime rates than others, a phenomenon which is subject to many crime theories and studies. Clusters of crime are referred to as hot spots. They describe areas, where a higher than average number of offenses occurs [Eck+05]. Hot spot policing focuses law enforcement resources on those clusters of increased crime. The effectivity of this strategy is subject to several studies, which indeed provide evidence that it is able to reduce crime, instead of merely relocating the places of crime [BPH12] [Wei+17].

This master thesis is researched in the context of the joint project „ICT4DMZ – Information and Communications Technology for Development in Mozambique“ of the TU Wien and the Eduardo Mondlane University (UEM) in Maputo, Mozambique. The relation to this research project provided the object of study for this work. Two research questions are formulated and investigated:

> *RQ 1: Is the routine activity theory, which assumes that crime rates decrease when there is more time spent at safe places, valid for an agent-based street crime model representing Maputo City?*

> *RQ 2: Are law enforcement officers, who patrol along the crime hot spots of the standard model, significantly reducing the crime rates in the advanced model? To what extent does relocation of crime to new clusters occur?*

1.2 Motivation

In 2014, the TU Wien, the Eduardo Mondlane University in Maputo, Mozambique and the nonprofit organization ICT4D.at set up a sustainable research partnership between the two universities. One of the project objectives is the realization of two research projects by students from the TU Wien in cooperation with students from the Eduardo Mondlane University in the area software engineering and geographic information systems [Gre13].

In project-related workshops, students from the UEM voiced their concerns, that some districts of Maputo are not generally safe in terms of street crime and that they desired to apply a GIS to investigate patterns of street crime as their student project. The topic „Street Crime in Maputo“ was not authorized by the University, so the students developed an Android app for students to find points of interest (POIs) like lecture rooms, Wi-Fi hot spots or public power plugs at the UEM campus. In this situation, one of the two research studies from the TU Wien was planned with the working title „Street Crime in Maputo“ instead. The present thesis is the result of this study of a GIS-related agent-based simulation in the context of the City of Maputo in Mozambique. The selection of a capital city of a developing country for evaluating the routine activity theory is a new approach, which is expected to lead to new results applicable to similar cities.

The Republic of Mozambique is situated in the south-eastern part of Africa and covers an area of 799 380 square kilometers. The country was a Portuguese colony from the fifteenth century until it attained political independence in 1975. The capital, Maputo City, has a population of 1.104 Million in 2020, or comprises about 6.1 % of the total population of Mozambique [Uni20]. The crime statistics by the Instituto Nacional de Estatística – Moçambique reports 5828 street robberies for Maputo City in the year 2015 [Ins15].

Computer simulation is a suitable method for exploring problems in the area of social sciences. The reason is, that for investigating a social phenomenon there often exists a lack of empirical data at the individual level. For example, privacy concerns and ethical considerations are involved when the study includes recording daily activities of humans. This issue of not being able to acquire enough empiric data at individual level, but only at macro level is related to the *ecological inference fallacy*. It occurs when statistical data about groups of individuals is misinterpreted to also be valid for the individuals. Agent-based simulations circumvent this issue by a different approach. General empirical data from the society-level and agent rules based on human behavior are the basis of the simulation model. The agent-based social computer simulation generates result data about daily activities of humans, from which conclusions about the target systems can be obtained.

The combination of GIS and ABS should contribute to a more realistic setting of the simulation. A model, which is closer to its target, should provide deeper insights and it is supposed to deliver more accurate results.

1.3 Expected Results

The first goal of the work is to evaluate the routine activity theory by means of a computer simulation based on real geographic data, census data and crime statistics from the City of Maputo in Mozambique. The second goal of the work is to investigate the efficiency of hot spot policing. The scientific goal of the computer simulation is to gain a greater insight rather than prediction. The work should contribute to a better understanding of how the social aspects, time and space dimension of human activity and interaction pose an influence on criminal events.

The type of criminal acts under study are direct-contact predatory violations, or street robbery. Other types of crime are not modeled. Street robbery is classified as an instrumental crime. This type of criminal acts strives for economic gain and therefore are a result of rational choice rather than an expressive crime. Human behavior resulting from rational choices can be modeled by simple agent rules. The study does not aim to predict criminal events, but to test if the behavior of the routine activity theory is reproduced. The evaluation of both hypotheses is accomplished by developing an agent-based model that is capable of three main features: Reproducing the convergence in space and time of the three elements of crime, the routine activity spaces of citizens and the human interactions that lead to a street robbery.

The study consists of several parts. In the first part, the relevant theoretical background is researched and discussed. This research concerns the areas of geographic information systems, computer simulation with a special focus on agent-based models and the routine activity theory. The research on GIS should provide a profound foundation for integrating real mapping data into a computer simulation and to realize the routine activity spaces of citizens. It should also include the tools for further studies such as spatial analysis of crime, which is not within the scope of this work. The research on computer simulation should provide an overview of simulation approaches and methods, a discussion about the scientific applications of computer simulation and a proposed framework for establishing trust in a computer simulation. The results of a computer simulation only have scientific value if the methods of verification and validation (V&V) are part of the model development process. The research on the theoretical foundation concludes with the presentation and assessment of the routine activity theory by Cohen and Felson.

Since this work researched is in the context of computer science, the focus lies within the development of an agent-based model, the integration of GIS and computer simulation, the correctness of the implementation and the application of statistical methods for theory testing. The challenges of programming this agent-based model are the realization of agent movement, and the imitation of basic human decision-making and behavior. The complexity of the simulation and the underlying model increases when this environment is simulated based on real-world geographic information system (GIS) data. The aim of the work is not to build realistic human agents with advanced techniques from the area of artificial intelligence. The work only uses very basic concepts of sociology and criminology. A deeper study of these aspects is not within the scope of this work.

The practical part of the work consists of building an agent-based model called *Street Crime Model* for simulating crime in an urban setting. The study target is the City of Maputo. Citizen agents should stay at home for a defined amount of time and then move around the city, according to their routine activity spaces. Human interaction should be imitated as street robbery involving a motivated offender, a suitable victim and the absence of a guardian. The simple human interaction is achieved by defining agent rules. In order to draw valid conclusions from the model, the parameters are based on empiric studies, census data and statistics.

The Street Crime Model is implemented in a suitable simulation software tool in order to investigate the routine activity theory by Cohen and Felson, which states that crime has an inherent geographical quality. At the beginning of the study, no simulation package, which supports agent-based simulation of street crime within realistic representations of real cities, was available. The solution is a middle ware approach. The software Agent Analyst integrates a GIS and an ABS toolkit. The city map of Maputo as a network of streets should serve as the environment of the simulation. Each citizen agent should have its personal activity spaces and the according route that connects those locations. While citizens visit their activity spaces, police officers move randomly. In addition to this standard model, an advanced version of the model is implemented, where the movement of police officers is based on crime hot spots. The simulation model should be tested, verified and validated using best practice techniques in order to make sure that the results are valid. A sensitivity analysis is used to calibrate the model and to increase the validity of the simulation.

The simulation should produce results at agent level and aggregate results for every simulated day. Results are recorded for citizens, police officers, places and robbery events. This generated data is used for the evaluation of the proposed hypotheses. In order to evaluate hypothesis one, twelve scenarios are simulated, each with a systematically changed percentage of the day spent in safe places input parameter. The influence of the percentage of the day spent in safe places on the number of committed offenses is investigated by a linear regression model. The second hypothesis is examined based on data of place agents and comparing crime rates of hot spot places of both the basic and the hot spot model.

The findings of this work are applicable to cities that are comparable to the city of Maputo. The applicability of results to other cities depends on the main demographic attributes that serve as parameters of this model. The parameters are population, unemployment rate, income, wealth and crime rate of street robberies. The city street layout is also a factor to be considered, because it influences the spatial distribution of crime and the convergence of offender and victim. A higher similarity in terms of the presented parameters will lead to stronger applicability of the results. For example, the following African cities are very similar in terms of city proper population (±100 000): Kigali (Rwanda), Fez (Morocco), Bujumbura (Burundi), Freetown (Sierra Leone), Monrovia (Liberia), Khartoum North (Sudan), Port Harcourt (Nigeria), Lilongwe (Malawi), Niamey (Niger), Nouakchott (Mauritania), N'Djamena (Chad) and Tangier (Morocco) [Wik21].

This work builds on existing contributions (see Chapter 3 Context and Related Work) to the topics of routine activity theory and agent-based simulation and investigates the subject using an improved model in terms of agent decision making, agent movement and modeling of the environment. A more detailed model produces more accurate data and provides more trust with regards to the evaluation of the results. Another novelty of this approach is that crime hot spots and different policing strategies are evaluated relating to the routine activity theory. The evaluation of the routine activity theory combined with an African capital city as an object of study is a new approach and should lead to

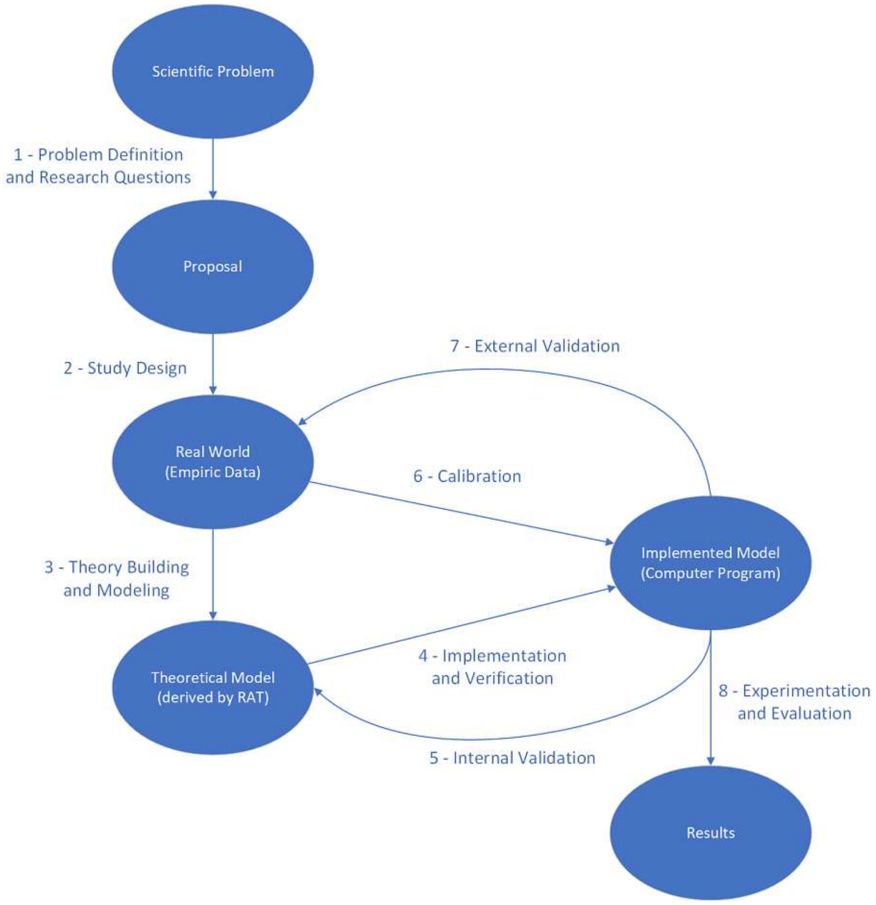

Scientific Problem
1 - Problem Definition and Research Questions
Proposal
2 - Study Design
7 - External Validation
Real World (Empiric Data)
6 - Calibration
Implemented Model (Computer Program)
3 - Theory Building and Modeling
Theoretical Model (derived by RAT)
4 - Implementation and Verification
8 - Experimentation and Evaluation
5 - Internal Validation
Results

Figure 1.1: The methodological approach as a process. Adapted from [HT15], [GT05], [Ül+06].

new scientific findings.

The practical relevance of the results exists for several aspects. If the propositions of the routine activity are evaluated to be valid for the object of study, the applications and implications of the routine activity theory (see Chapter 2.3.2) also apply to Maputo and similar cities. The main consequence would be to avoid the convergence of the three elements of crime. If only one of the three elements of crime is missing, no crime takes place. Crime prevention strategies can target each of the three elements. This work provides a scientific foundation for city government and law enforcement organizations to evaluate and refine their crime prevention policies.

1.4 Methodological Approach

This work is an interdisciplinary study with focus on computer science. The research of the theoretical background concerns the areas of geographic information systems, computer simulation and the routine activity theory [CF79]. This socio-behavioral theory describes the occurrence of crime in a spatio-temporal context. The scientific method of simulation is a design science method, in which the built artifact is executed with artificial data [Hev+04]. In this study, an agent-based computer simulation is applied for theory testing [GT00].

The methodological approach of this study and the implemented general framework of the modeling and simulation process are presented in Figure 1.1. The presented approach is based on the works of Happach and Tilebein [HT15], Davis [DEB07] and Ulgen et al. [Ül+06]. In the literature, simulation is regarded as a scientific method that is performed in steps or phases with iterations.

The first step of the process addresses the problem definition and the formulation of the research questions [HT15]. Then, the methodological approach and the expected results are defined and the findings of the first step are documented in the proposal. Step two, the study design, puts an emphasis on technical aspects of the study. Results from the previous step are examined in more detail. Simulation software and required tools are selected. The required empirical data for parameterization and validation is identified and researched [Ül+06].

The third step applies the method of modeling to deduct a simplified representation of the reality. The study of the routine activity theory results in a theoretical simulation model including agent-level behavior of daily routines and street robbery. The practical aspect of this work is described in step four, which describes the implementation of a computer program based on the specifications provided by the theoretical model. The computer program simulates interactions of citizens in a spatio-temporal environment representing the city of Maputo, Mozambique. This step also includes the verification process, which is responsible for ensuring a high degree of software correctness.

The validation steps of the simulation aim to ensure the credibility of the results [Bal94] [Klu08]. In step five, simulation results are compared to the theoretical model and it is verified, that the simulation works as the street robbery model has been specified. The internal validity aims to secure the cause-effect relation between the dependent and the independent variables in the simulation. In step six, input parameters of the simulation are selected from empiric research data. This task describes the calibration of the model.

The external validation step seven compares simulation results to the real world system under study [HT15]. It is tested whether the simulation is able to reproduce empiric data. In order to achieve external validation, two variables from empiric research are investigated. In this work, the time spent in safe places is treated as the exogenous variable and the resulting crime rate is treated as the endogenous variable. The time spent in safe places is used as input parameter and the simulation should reproduce the correspondent crime rate. The sensitivity analysis aims to enhance the robustness of the model and comprises of changing parameters of agents and their environment and results in different output patterns of street robbery [Klu08].

Once the validity of the model is established, the street robbery model can be applied to investigate and explain the propositions of the routine activity theory. This is achieved during the experimentation step eight [HT15] [Ül+06]. The goal of the simulation is the evaluation of the two proposed hypotheses. Experiments are conducted to test if the implemented model is able to reproduce the predicted behavior from the routine activity theory. In order to draw valid conclusions from the result data and to prove or to refute the research questions, statistical methods such as linear regression and statistical tests are performed.

1.5 Structure of the Work

Chapter 1 provides an introduction to the topic, the problem statement and the motivation for the topic. It formulates the expected results and explains the selected methodological approach. Chapter 2 is concerned about the theoretical background of the scientific fields relevant to the work. It provides definitions and an overview of geographic information systems, computer simulations in research and science and the routine activity theory by Cohen and Felson. These basics serve as background for the practical tasks of formulating a spatially-aware agent-based model, testing and implementing the model, running the simulation and interpreting the results. Chapter 3 provides an overview of the scientific context of this work and present related research. The focus lies on agent-based simulations with a geographic component for crime prevention and analysis. The simulation model „Street Crime Model“ is presented and discussed in Chapter 4. All involved agents and the environment as the main constructs are described. Furthermore, agent rules are defined, which enable basic interactions and decision-making. The parameters of the model are listed and discussed. In Chapter 5, the implementation of the previously presented Street Crime Model and the development of the simulation program is described. The results of the simulation research are presented in Chapter 6. First, general model results and descriptive analysis based on the standard model are provided. Then, the results of the advanced model are assessed. In the last part of result section, the proposed two research questions are evaluated based on the result data. Finally, Chapter 7 provides a summary of the relevant findings of this work, lists limitations of the research and gives an outlook of possible work in the future.

CHAPTER 2

Foundations

This chapter presents the foundations and concepts of a spatially-aware agent-based social simulation. The spatial dimension of the simulation is represented by a geographic information system. The first part of the foundations chapter provides a comprehensive overview of geographic information systems with a focus on those aspects that contributed to this study. The modeling and simulation dimension is covered by the second part of the foundations chapter. Computer simulation is introduced as a relevant scientific method, different modeling and simulation approaches are discussed and the crucial process of establishing model credibility is elaborated. The simulation model of this study is derived from the routine activity theory [CF79], which describes the elements of crime in a spatio-temporal context. This socio-behavioral theory and its implications on the simulation model are discussed in the third part of this chapter.

2.1 Geographic Information Systems

The first section provides an introduction to geographic information systems, a brief history and further characterization of the subject (Section 2.1.1). An overview of GIS applications in various settings is presented in Section 2.1.2. The science behind geographic data is discussed in the following Chapter 2.1.3. The shapefile data format plays an important role in storing and exchanging spatial data within this study (Chapter 2.1.4). Techniques and trends of spatial visualization are explained in Chapter 2.1.5. Information about GIS related software is presented in Chapter 2.1.6 with a special emphasis on Esri ArcGIS, which is used in this study.

2.1.1 Introduction

Geographic information systems is a discipline that studies the „description, explanation, and prediction of patterns and processes at geographic scales“ [Lon+15]. The Oxford

English Dictionary defines geographic information system as an „information system which allows the user to analyze, display, and manipulate spatial data, such as from surveying and remote sensing, typically in the production of maps; abbreviated GIS“ [Oxf16].

The term geographic information systems connects several disciplines. Geography describes the science of the description of the Earth's surface including countries, landmarks, seas, mountains, towns. An information system is „an integrated set of components for collecting, storing, and processing data and for providing information“ [The16]. Information systems in the scientific field of geography serve for capture, storage, analysis and visualization of data, describing a subset of the Earth's surface and its phenomena. While the terms *information* and *data* are often used interchangeably, there are certain restraints for data in information science. In contrast to data, information is structured and it is enriched by semantics and relevance [Bar95]. An adjective with almost synonymous meaning to geographic is *spatial*. Spatial is not bound to the Earth's surface, but may refer to any space [Lon+15]. In the literature, one can observe the use of the term *geospatial* instead of the traditional term *geographic*. Caitlin Dempsey Morais from GIS Lounge[1] sees *geospatial* as a broader term, referring to all possible applications and technologies of geographic data [Cai12].

Two more definitions of GIS from leading institutions show the range of valid interpretations for this term. Esri, software development company of the market leading GIS software ArcGIS, provides the following definition [Env16b]:

> „A geographic information system (GIS) lets us visualize, question, analyze, and interpret data to understand relationships, patterns, and trends.“

The United States Geological Survey (USGS), a U.S. government research organization, published this interpretation [Uni16]:

> „A GIS is a computer system capable of capturing, storing, analyzing, and displaying geographically referenced information; that is, data identified according to location. Practitioners also define a GIS as including the procedures, operating personnel, and spatial data that go into the system.“

The Three Dimensions of GIS

In this thesis, GIS is used as scientific methodology for discovery of new knowledge, reproduction of objective experimentation results, method of validation and problem solving. In order to classify a scientific GIS related problem, the literature proposes three dimensions.

The first property is *scale*. A digital representation of a real-world entity is a model, or more specific, a conceptual model. It is formed after a generalization and abstraction

[1] https://www.gislounge.com (visited on 03/19/2021)

process. The goal is to formulate an abstraction of the reality while conserving only relevant attributes. Geographic models usually show artifacts at a lower level of detail for practical reasons, for example a simplified mountain on a map at a 1:24,000 scale. Second, GIS problems can be categorized by their *purpose*. Some are driven by economic goals, others by practical needs, or, when GIS is used to verify a scientific thesis, to advance the human understanding of the world. These very different intents do not necessarily use different tools and methods to achieve their goals. Third, next to the dimension of scale and purpose there is also a *time* dimension. Geographic data requires a temporal component in order to be able to describe the change of entities over the passing of time [Lon+15].

A Brief History of GIS

Historically, GIS was driven by the digitalization of simple geographic measures. The term *geographic information system* was first used by Dr. Roger F. Tomlinson in his paper *A Geographic Information System for Regional Planning* [Tom69] during his directing work at the Canadian Geographic Information System (CGIS) [Env16a]. CGIS was a governmental initiative to analyze Canada's land resources in the 1960s. The U.S. Bureau of Census implemented a database of all streets in the U.S. in order to support the 1970 Census of Population [CR91]. The first general purpose GIS software was ODYSSEY GIS by Harvard University's Laboratory for Computer Graphics and Spatial Analysis [Chr06] [Tei80]. At this time the primary goal was to lower the required time and costs for map creation. The first computer-created map as shown in Figure 2.1 was published in 1971 by the UK Experimental Cartography Unit (ECU).

Digital remote sensing, first applied by military and later by civil institutions, fueled the development of GIS. Digital satellite systems such as the NASA/USCS joint project *Landsat*[2] [NAS16] [Sho82] provided massive data for GIS and the Global Positioning System (GPS) [ME06] revolutionized location based services, making it possible to measure the position on the Earth's surface. The modern age of GIS started in the 1980s when computing hardware prices declined and a related software industry developed.

Components of a GIS

In the litarature, definitions of the components of a GIS vary dependent on the author's intended level of detail, but they have four components in common: hardware, software, data and people [Bar95]. Paul A. Longley et al. expand this view with the network, which enables communication and dissemination of information between the other five parts as shown in Figure 2.2. Today, the paramount network is the internet, connecting billions of people, personal computers, servers and other devices. GIS is deeply integrated in the internet and increasingly taking advantage of wireless networks and mobile devices. Presently, mobile devices offer real-time location-based services such as maps, navigation, transport information and location aware search. Application programming interfaces

[2] `https://www.usgs.gov/land-resources/nli/landsat` (visited on 03/19/2021)

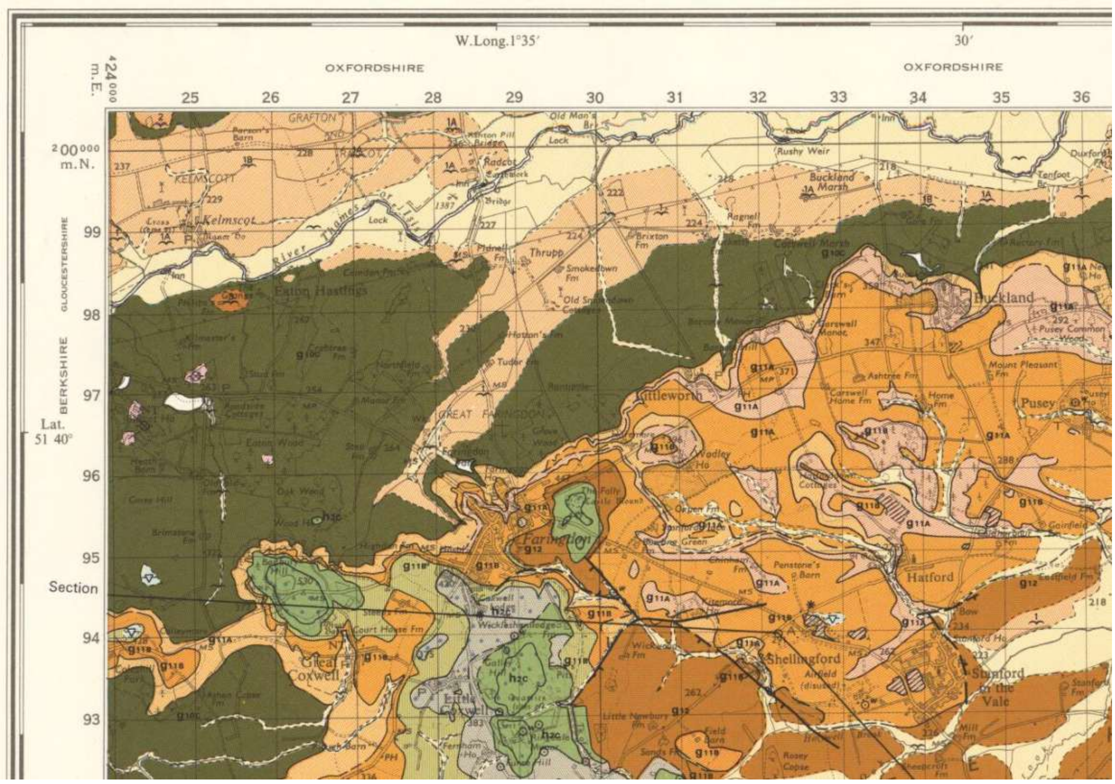


Figure 2.1: Section of the Geological Survey of England and Wales 1:63 360. Published 1971. Experimental Production. Automatic Cartography by the Experimental Cartography Unit. Contains British Geological Survey materials © NERC 2016 [Bri16].

(APIs) are provided by many web GIS applications (Google Maps[3], Microsoft's Bing Maps[4], OpenStreetMap[5], Foursquare[6] etc.) enabling users to collaborate and participate at creating custom online GIS services [Lon+15].

The second part of a GIS is the computer hardware. It is the device the user interacts with, it carries out procedures and algorithms on data, and stores input and output information. GIS hardware typically consists of a client-server architecture. The software that runs on computer hardware is considered another part of the GIS anotomy. It can range from a simple web browser on a (thin) client which is connected to a remote server offering GIS related services, to dedicted GIS software packages from a GIS vendor, open-source or academic project. Section 2.1.6 covers GIS software evolution, architecture und products in greater detail. GIS Software usually operates on a database, the fourth

[3] https://developers.google.com/maps (visited on 03/19/2021)

[4] https://www.microsoft.com/maps/choose-your-bing-maps-API.aspx (visited on 03/19/2021)

[5] https://wiki.openstreetmap.org/wiki/API (visited on 03/19/2021)

[6] https://developer.foursquare.com (visited on 03/19/2021)

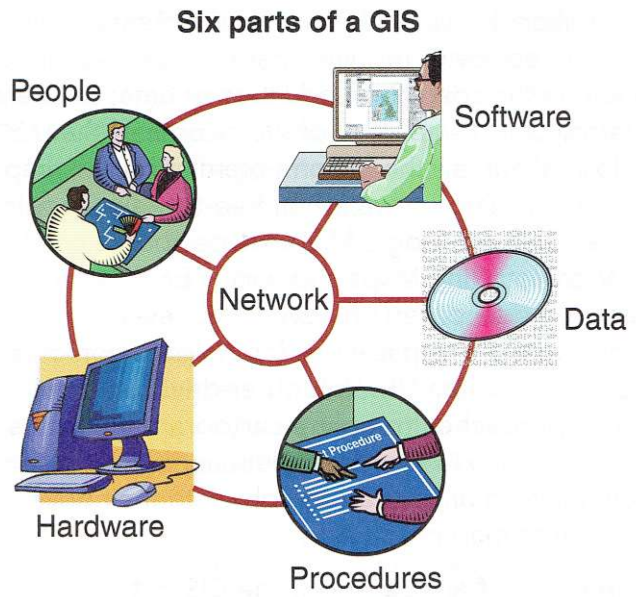


Figure 2.2: The six components of a geographic information system [Lon+15].

part of a GIS. The GIS data stored within is a digital representation of a model of the Earth's surface. The size of such a database may range from a few megabytes (street map of a small city) to petabytes (satellite imagery of the entire Earth's surface at one meter resolution). Section 2.1.3 provides an overview of selected attributes and types of geographic data. Finally, all mentioned components would be without purpose without people to use, maintain and develop geographic information systems [Lon+15].

2.1.2 GIS Applications

Geographic information systems affect our everyday life. GIS are of enormous practical importance and can be applied to solve socio-economic problems. They enable effective decision-making, generate measurable benefits to society and economy and they grant us the so-called *5 Ms of GIS*: mapping, measurement, monitoring, modeling and management of the environment. In the following paragraphs, five key-areas of GIS applications are introduced and described: science and GIS operations, government and public organizations, business and economic appliances, transportation and environmental applications [Lon+15].

Scientific GIS Applications

One application of GIS is solving complex *scientific questions* and models that represent real-world problems. GIS adds the potential of the spatial domain to science, enabling problem analysis and solving in a wide range of scientific areas. For example in economics, one goal is the effective allocation of resources. Another field of study is the observation of certain geographic attributes distributed in a specified region. In social sciences, one application is the analysis of the difference that place constitutes for human characteristics and behavior. For example investigating regional varations in people's dialect or surnames, or examining voting habits that depend on the place of living. In geology, one research area is to understand processes and their changes over time of natural environments like land erosion, water distribution or river delta deposition. In an effort to grasp and solve these research questions, data manipulation operations such as geographic database techniques, spatial mapping and inventory compilation are required [Lon+15].

In the literature *geographic information science* or *GIScience*, a term first used in a paper published by Michael Goodchild in 1992, studies the fundamental problems emerging from georgaphic information. GIScience is a relevant field of study, proved by several renamed scientific journals (see Table 2.1) and research facilities. The most relevant organization is the US University Consortium for Geographic Information Science (UCGIS[7]), an alliance of 54 universities and the two affiliate members Esri, Inc and US Geological Survey. In addition to an ambitious research agenda, educational goals and building research networks, UCGIS is lobbying for open access and ethical use of geographic information and technologies. An international GIS conference *GIScience*[8] is held biannually since 2000. Another notable international non-profit organization in the field of GIS is the Open Geospatial Consortium (OCG[9]), founded 1994 with the goal to develop and establish open standards for geospatial data. Today, the OCG is comprised of more than 500 commercial, governmental, non-profit and research organizations [Goo92] [Lon+15] [Uni12].

Governmental GIS Applications

The second area of GIS applications is the *public service and government* sector. As already discussed in Chapter 2.1.1, goverments were among the first users to utilize geographic information systems. Today, public organizations still form the biggest group of GIS professionals, depending on GIS to support decision-making at all levels from a local scale to the whole nation. The most relevant use cases for governments are in the fields of natural resources, infrastructure, transportation, land development, public services and administration. According to O'Looney [O'L00], GIS applications can be classified in inventory applications, policy analysis applications and management/policy-making applications. Inventory applications focus on the management of property information and infrastructure, for example a cadastre of housing, or inventories of roads, water pipes, rail tracks, police stations, land uses or social risk indicators. While inventory applications have the purpose of monitoring and mapping, policy analysis applications apply mathematical and statistical operations to geographic data in order to gain additional information. Essentially, the simulation model developed in this master thesis can be categorized as policy analysis application, because its simulation results can support policy-makers to decide law enforcement related policies, like for example police

[7] `https://www.ucgis.org` (visited on 03/19/2021)
[8] `https://giscience.org` (visited on 03/19/2021)
[9] `https://www.opengeospatial.org` (visited on 03/19/2021)

Scientific journals with focus on GIS research
Annals of the Association of American Geographers
Cartography and Geographic Information Science
Cartography – The Journal
Computers and Geosciences
Computers, Environment and Urban Systems
Geographical Analysis
GeoInformatica
International Journal of Geographical Information Science
ISPRS Journal of Photogrammetry and Remote Sensing
Journal of Geographical Systems
Photogrammetric Engineering and Remote Sensing
Terra Forum
The Photogrammetric Record
Transactions in GIS
URISA Journal

Table 2.1: Scientific journals with focus on GIS research. Adapted from [Lon+15].

visibility and presence, the number of law enforcement officers in relation to density of criminal activity, or victim profiles compared to residental populations. The third class of governmental GIS applications, management/policy-making applications, consist of decision-support systems for providing information, evaluation, analysis and forecasts. Two examples are the evaluation of a land-use policy based on demographic and social attributes of the residing population, or an application for emergency management that places relief units and facilities at the most efficient locations [Lon+15] [O'L00] [Paw01].

GIS Applications in the Industry

The next area of GIS applications to be discussed is the *business and industry* sector. This field utilizes spatial data and operations to attain context for decision-support on the basis of their business strategy. For example, spatial and social attributes such as consumer behavior at a regional level, called *geodemographics*, are of interest for market researchers. Their activities of examining the relationship between retail facilities and density of potential customers is described by the term *market area analysis*. This method can also be utilized as a blueprint for non-profit or governmental GIS applications such as improving health, education and law enforcement services to the public. Geographic information systems may serve the *operational*, *tactical* and *strategic* goals of an enterprise to reflect its day-to-day, short term and long term needs. These software packages may range from standard GIS software products like Esri's ArcGIS to custom-developed, sophisticated business software. *Operational* GIS applications serve the purpose of computing daily routine transactions and algorithms, for example warehouse management. Some of these operational fuctions such as distribution networks or stock flow management systems are discussed in the next paragraph in the context of transportation and logistics. These areas are very tightly connected to the business and industry sector. *Tactical* applications operate on spatial and business data to solve specific problems, for example quarterly sales promotions or marketing campaigns. The long-term objectives and mission of an enterprise are supported by *strategic* GIS applications to cover problem statements such as expanding or rationalizing the retailing network. One important scientific principle is called the Tobler's First Law of Geography, stating that „everything is related to everything else, but near things are more related than distant things" [Tob70]. A very important scientific question evolving from geodemographics is predicting the complex human decision-making by understanding local patterns [Lon+15] [Sui04] [Wat17].

GIS Applications in Transportation and Logistics

Closely related to businesses is the field of *transportation and logistics*, which engage in the commercial shipping of commodities and people from place A to B and the underlying infrastructure. Logistics companies have to deal with problems like the NP-complete travelling salesman problem [DG97], which is: given a list of waypoints and the distance between each of them, what is the shortest possible path visiting each waypoint exactly once and the first stop equals the last one? Another domains for GIS in this area are placement of warehouses and distribution facilites, or routing of parcels from origins to destinations, very often by different means of transportation: ship, truck and railway. Transportation enterprises like airlines, bus companies, railway companies or highway authorities have to face very similar challenges today: planning, evaluation and establishment of routes and schedules, micro-management and monitoring of vehicles and transportation systems in real-time and dealing with incidents and delays. All these presented businesses heavily rely on GIS and its previously presented applications of operational, tactical and strategic scope. In the early years of GIS, the transport applications centered around the static part, the infrastructure. But today, GPS technology enables live-tracking of single vehicles and transport service companies use information systems to inform their customers on live departure times and exact locations [Lon+15] [ME06].

Environmental GIS Applications

In order to conclude the overview on GIS applications, the environmental applications are discussed. They can be counted as the earliest application of GIS, implemented in the mid-1960s by the CGIS, a governmental initiative to analyze the Canada's land resources (see the historic overview given in Section 2.1.1) [Tom69] [Env16a]. In the present, remote sensing from space offers more effective applications like investigating land use change and urban growth. Increasing populations turn the focus in science and politics to acquiring an understanding of the environmental consequences of urban settlements. Researchers use GIS to investigate urban sprawl and formulate predictions based on patterns of expansion, together with spatial data, land use and accounting for zones that are uninhabitable such as streets, lakes and mountains. These variables may be represented each as a layer of a digital map in a GIS. Together with simulation software, this geographic data can be used to compute the processes of growth. Urban growth models are a classic application of *dynamic system simulation* [He+06] [MHS95] [WE93], which models the behavior of a system and its changes over time and *agent-based simulation* [AHNV13] [LMOM12] [LT03] (see Section 2.2 for more detailed information on simulations). More environmental applications of note are groundwater analysis, soil erosion or forest growth [Lon+15].

2.1.3 Geographic Data

Geographic data consists of three elements: place, time and an attribute component. The third element could be temperature, elevation or even unemployment. Some attribute components are natural, other physical, geological, social or economic. There are identifying attributes like postal codes and street numbers, measuring attributes like precipation and classifying attributes for differentiation between nations, land use or terrains. A categorization of these attributes results in the following five types: *Nominal* attributes such as street numbers only identify or distinguish one entity from others. Mathematic operations applied to nominal entities do not result in any useful data. Attributes are of *ordinal* type if their values follow a natural order. For example lakes might be rated in classes by water quality. Calculating the median is a common operation with ordinal types. When differences between values matter, the attribute is of *interval* type like the Celsius temperature scale. When ratios between values matter, the attribute is of *ratio* type like weights or heights. The last category is cyclic data, including compass directions, longitude/latitude, month of the year, or degrees where the number following 359 is 0 [Lon+15].

Because the world is infinitely complex, for example shorelines keep revealing new details when magnified until the size of subatomic particles, a representation of the world must limit the level of detail. The principle of modeling is to omit information of the reality that is not relevant for the viewer. Similarly, cartographers as well as GIS are limiting detail by working with a *spatial resolution*, which describes the smallest possible feature that is detectable. Depending on the *scale* this feature could be a tree or a small river or an entire country [Gov08].

Beside the level of detail, another relevant aspect of representation is shared by two concepts. These basic views of representing geographic information are discrete objects and continuous fields. The first concept assumes the geographic world as empty space filled with a finite number of objects. These objects have defined boundaries and can be seen as instances of categories. Discrete objects are classified by their dimensionality. Zero-dimensional objects are infinitely small and represent points like buildings, trees or other entities of interest. An artificial point reference is the *centroid* of an area object, located at the focal point in order to provide a summery attribute for the object. One-dimensional objects are recognized as lines, defining borders, roads or rivers. Lines have length, but no breadth or depth, so they can be used to calculate distances between spatial objects. Two-dimensional objects are identified as areas or polygons. They have

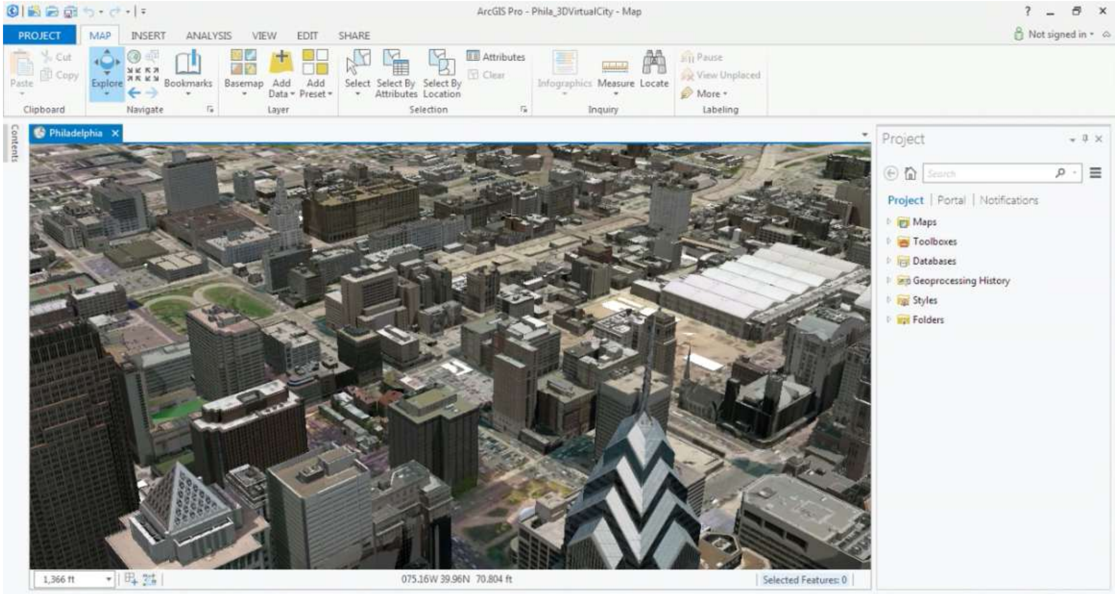


Figure 2.3: A digital representation of Philadelphia in 3D opened in the GIS Software Esri ArcGIS Pro [Che15].

length and breadth, but no depth. Areas are often used to represent natural objects such as lakes or forests, but are also used to represent artificial entities, such as borders or census tracts. In contrast to a paper map, modern GIS are able to process three-dimensional objects as shown in Figure 2.3. Objects of the third dimension have length, breadth and depth and represent natural entities such as mountains or artificial objects like population density. Sometimes, time is considered the fourth dimension of spatial objects, although GIS visualizations are poorly suited for the modeling of temporal flows [Lon+15].

The second concept regards the world as a continuous surface with a defined elevation at every point. These continuous fields, or nondiscrete data, describe the surface by variables which have a value that is measured from a fixed point, for example the sea level for elevation. Topographic maps are the most common type of a continuous field where elevation is represented by a color gradient. But also other geographic features can be described by continuous fields such as population density, traffic, or flow volume of a river. These presented concepts are not suitable to be implemented in a computer, because there is an infinite amount of information in any given geographic area. As a result, two methods have been developed in order to digitalize geographic information. Rasters and vectors are the two paramount methods for digital representation of geographic information. Although both methods can be used to work with discrete objects and continuous fields, discrete objects are usually represented by vectors and continuous fields by rasters [Lon+15].

Raster and Vector Data

In a *raster* representation, the world is grouped into an array of square cells, a set of rows and columns, much like a chess board. In order to represent geographic features, certain cells are assigned specials properties, for example a color. Like in image processing, these cells are also called picture elements or pixels. Figure 2.4 presents three different kinds of raster representations: point, line and polygon features. In reality, lines and polygons are rarely straight and rectangular and mostly curved, so rasters can only give an approximation of the original geographic feature, because each cell is assigned a single value. In order to represent a shoreline in raster data, rules must decide whether a pixel should get assigned a sea value or land value. One widespread rule is that the surface type with the largest share of the raster cell's area is being assigned. Another, computationally fast rule assigns the whole cell based on the attribute at the exact center of the raster cell. Raster models work well with the layer concept. The overlay of different raster layers is called composite map. Algorithms from image processing can be applied on composite maps [Bar95].

The raster data model uses the array data structure to store its information. Each element of the array can hold any data type based on the type of the raster model: boolean data type for binary representations such as land or sea, integer or floating-point values for topographic maps or multiple attributes in a nested array. This data is stored as an array of grid values including metadata in the file header. Metadata is usually composed of the geographic coordinate of the upper-left corner of the grid, the number of rows and columns and the projection (see Section 2.1.5 on projections).

While the raster data concept is rather simple, the amount of data generated can be too large for computer systems to store and process. This is especially the case for remote sensing raster imagery from satellites, which is usually described in RGB values. Assuming these values range vom 0 to 255, each pixel requires 3 Bytes, so a 2000 x 2000 pixel map results in over 11 Megabytes of data. As a result, there are several compression techniques in order to reduce the amount of data needed to store raster data layers [Wis13]. *Run-length encoding* [Teu78] is a simple and lossless encoding algorithm for clustered, non-random data. It counts adjacent row cells with the same values and substitutes them with pairs of count and single data value. Considering a binary raster map showing only land and sea in two colors. A section of the array, where L represents land and S represents sea might read as follows:

$$LLLLLLSSLLLLLSSSLLLLLLLLSSSSLLLLLLLLLL$$

With run-length encoding data compression applied to this input array, the resulting output is:

$$6L2S5L3S8L4S10L$$

Wavelet transform encoding [Ant+92] is widely used in geographic applications, especially for compression of satellite imagery, and has been implemented in the JPEG 2000 standard [Mar+00] for image compression. The algorithm first applies a wavelet function,

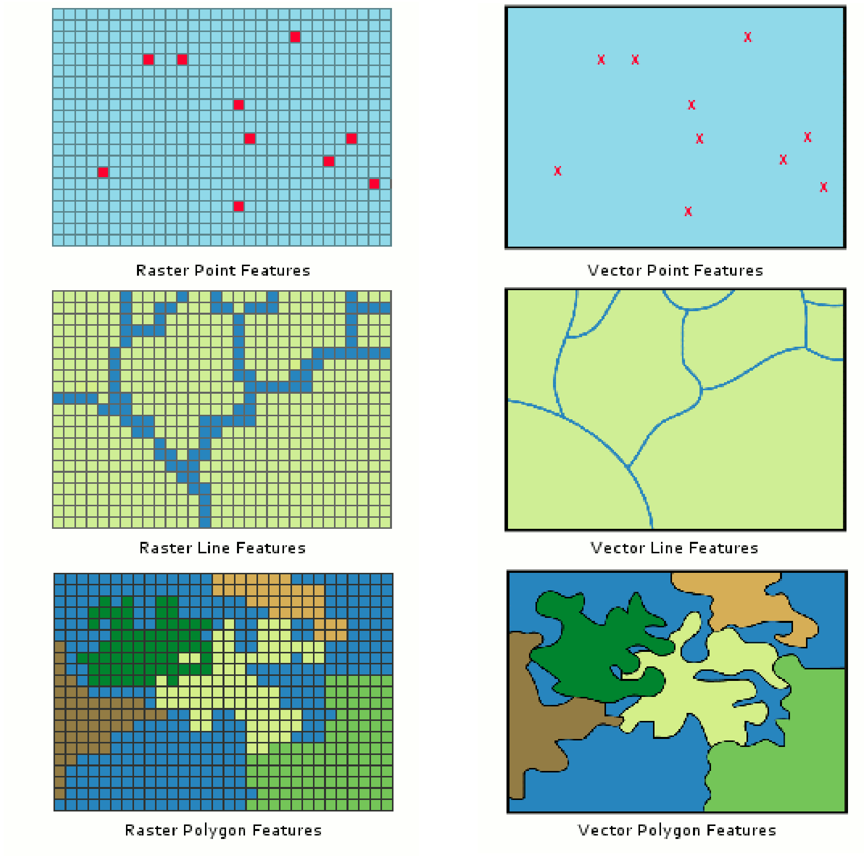


Figure 2.4: Computer representations of geographic features: raster and vector data. Adapted from [Her17].

transforming every pixel into a coefficient. These coefficients are easier to compress, because of statistical concentrations in only a few coefficients. In the second step the coefficients are quantized and the resulting values are run length and/or entropy encoded. Comparing these two presented algorithms when applied to an uncompressed original image, for example a topographical map showing shades elevation, the run-length method achieves a compression rate of about 5 and the more sophisticated wavelet compression achieves a compression rate of about 38 [Lon+15].

In a *vector* representation, all entities are made of points connected by straight lines. An area, or also called polygon, is circumscribed by vertices, which are connected by edges as illustrated in Figure 2.4. Curved geographical features such as lakes are usually approximated by increasing the amount of points, so they seem naturally curved for the viewer. Lines are captured similarly and a curved line described by a set of points connected by straight lines are called polyline as shown in Figure 2.4. An area object is described by capturing the points that form the outline of the polygon [Lon+15]. The location of a vertex can be described by a pair of *(x, y)* coordinates.

Adding a unique ID, a simple data structure of vector data is (ID, X, Y), respectively $(ID, N, X_1, Y_1 ... X_N, Y_N)$ for polylines. For Polygons the first point (X_1, Y_1) and the last point (X_N, Y_N) must be equal. This principle is used in the Keyhole Markup Language (KML)[10], standardized by the Open Geospatial Consortium. KML is an XML-based text format for storing geographical data and it was developed for use in the Google Earth application [Wer08]. Vector points in the form (ID, X, Y) can be stored in a multidimensional array POINTS[N,3], which corresponds to a grid with N rows and three columns. Using this simple data structure, basic computations such as finding the distance between two points are possible. However, there are limits and drawbacks to this approach. There is no information about which lines are connected, because each line is stored independently. As a result, it is not possible to plan routes along a network of lines using this data structure. When dealing with polygons rather than polylines, each line is stored twice, because the boundary is shared by two adjecent areas. The analysis whether objects are neighbours, *contiguity*, is part of the mathematical subject of graph theory. Topological data structures expand the aforementioned simple structures by topological rules and information and do not share their listed disadvantages. Additional information is stored in order to enable or accelerate certain calculations such as routing. The Dual Independent Map Encoding (DIME) [US 90] data structure scheme was the first topological data structure, developed by the US Bureau of the Census in 1965. There are two large disadvantages of topological data structures. In order to draw one object such as a polygon, the renderer must retrieve the necessary information from different files and tables. As a result, some simple and frequent operations are slowed down, especially with large datasets. Second, it is more complicated to transfer complex topological data between different systems, while it is trivial to store simple data structures in a text format. These issues resulted in the use of non-topological data structures for vector data. Current formats for storing vector data are for example Esri's shapefile format [Esr98], which is used in the practical part of this thesis and further explained in Section 2.1.4, and the already mentioned KML standard [PC75] [Wis13].

Raster and *vector* representation methods have been presented. Their use for certain applications depends on a variety of factors. Raster representations have their advantages for describing remote sensing and satellite imagery for resource-related or environmental applications. Vector data models are best applied for display of lines and polygons for adminstrative, social or economic applications. Raster models offer less accuracy than vectors, or take up significantly more storage space. Some calculations are much faster using vector models, for example transformation of coordinates. Other operations like finding neigbours or interpolations are trivial when used on raster data. There is also the concept of a *hybrid model*, aiming to combine the advantages of both rasters and vectors [Bar95] [Win98].

[10] KML documentation online at `https://developers.google.com/kml/documentation/kmlreference?csw=1` (visited on 03/19/2021)

Spatial Data Analysis

One important application of geographic data is analysis based on location. It is the challenge of comparing various attributes of certain locations in order to discover correlations and develop predictions. The previously presented concepts of *simple data* and *topological data* models emphasize on efficiency in representation and rendering. In these two models the attribute values for all shapes are stored in a single attribute file or table. This is called horizontal integration of data, while vertical integration of data would mean that all attributes and their corresponding values are stored in the Shape. For example, the elevation data for a federal state is stored in one place and the rainfall data of the same state is stored in another place. These data structures are not suited to analyze scientific questions like *Is there a relationship between elevation and rainfall in this federal state?* on a point-by-point basis. One method of spatial data research is analysis of attribute tables, which is comparing the values of two or more columns in order to discover possible relationships or correlations. A graphical method is to plot one variable along the x-axis and the other variable along the y-axis as a *scatter plot* as displayed in Figure 2.5. From the field of statistics, scatter plots show the dependence of one variable on one or more independent variables. In the example illustrated in Figure 2.5, coffee price is the independent, or exogenous variable, and consumption the dependent, or endogenous variable. For the linear regression model including more than one independent variables this example can be formulated as

$$Y = f(X_1, X_2, X_3, ..., X_k)$$

where Y is endogenous and X_1 through X_k are exogenous and might be correlated with Y. The exogenous variables together predict Y, but the reverse inference is not valid. The proposed equation is an approximation and does not explain all factors that are responsible for the resulting variable Y, so an error term ϵ is added:

$$Y = b_0 + b_1X_1 + b_2X_2 + b3X_3 + ... + b_kX_k + \epsilon$$

b_0 is the intercept term and $b_1...b_k$ are called regression parameters and define the direction and weight of the influence of the exogenous variables $X_1...X_k$. The slope of the line is computed as the b coefficient of the regression, whose sign indicates a positive or negative trend [AR15] [EKD11] [Lon+15] [Win07].

In Chapter 6.3, regression analysis is applied to the simulation results in order to find significant correlations and to answer the proposed research question.

Problems Emerging from Spatial Analysis

The relationships across space that can be uncovered by regression models are related to the issue of *spatial heterogeneity*, or uneven distribution of various biological, geological, or environmental entities within an area. This topic is further investigated in conjunction with visualization of geographic data in Chapter 2.1.5. Another interesting problem in

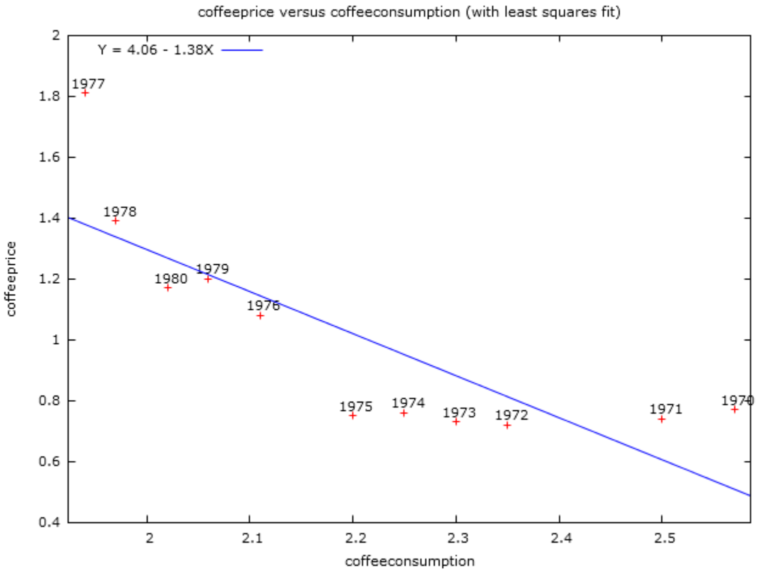


Figure 2.5: Example scatterplot of coffee price versus consumption using *gretl* (Gnu Regression, Econometrics and Time-series Library [All17]).

the area of spatial regression analysis is the *modifiable areal unit problem* (MAUP). This phenomenon was initially described by Openshaw in 1981 [OO81], where he illustrated that applying artificial units of spatial reporting, such as political districts, on continuous geographical attributes results in biased data. MAUP can occur in two types: Scale MAUP refers to the generation of different results when information at different scales like country, federal state, district, municipality is investigated. Hence, the scale must be matched to the research question. Zone MAUP refers to the issue of grouping schemes used for spatial analysis, even if the chosen units are of the same scale. Applying different geographic zones to the same set of data points yeilds different results. In politics, this phenomenon is called *gerrymandering* [Joh02], the manipulation of electoral district borders for political gain. In order to avoid this problem, zones should be chosen based on a comprehensible logic, such as simple shapes of equal areas or following phyisical, geographical or social divisions [JW96] [Won09].

Spatial Joins

In order to conduct spatial analysis, required attributes are often distributed among different attribute tables. A relational join is a basic operation to combine the contents of two tables using a shared unique key. One of the most important features of a GIS is the method to join tables based on common geographic location. For example, an analysis of the crime rates of different cities based on a GIS database where cities are stored as point shapes should be performed. Relevant attributes of the investigated cities are stored in attribute tables, like the endogenous variable monthly income and the exogenous variable level of education. Other potentially relevant attributes such as unemployment rate are only available at the state level. The identifier to join tables *city A* and *state* is *state*, which exists in every city record. The resulting table now includes the unemployment rate at the city level [Lon+15].

2.1.4 The Shapefile Spatial Data Format

Shapefile is the quasi-standard for geographic vector data, developed by Esri, Inc. for use with their desktop GIS software ArcView. It is now supported by many open-source and commercial products and libraries. As already explained in Section 2.1.3, vector data is rather suited for representing discrete objects such as points, lines and polygon features, instead of continuous fields. The shapefile format does not contain the processing overhead of topological information that is information about the relationships among objects of the same map layer. As a result, it has advantages oder other data models, for example faster drawing speed and edit ability, because the information for one single object is not scattered among several source files. Like related, simple formats of geographic data, shapefiles maintain a connection between the information about the location and discrete objects (sets of (X, Y) - coordinates) to their geographic attributes in a one-to-one relationship [Hil07].

Shapefiles are made up of at least three files which need to be stored in the same directory: a main file, an index file, and a dBASE table. The main file (.shp) contains records of variable lengths, each representing a shape with a list of its vertices. The index file (.shx) contains offsets of the corresponding records from the beginning of the main file. The dBASE table file (.dbf) stores attributes with one record per feature. This one-to-one relationship between a discrete object and its attributes is based on the unique record number. Attribute records stored in the dBASE file have to be in the exact same order as records in the main file [Esr98].

The *main file* contains a header with fixed length, followed by object records of variable length. The object records in turn are made up of a header with fixed length followed by contents of variable length. The main file header's size is 100 bytes. Table 2.2 shows the entries in the file header with their byte position, value and type. The shape type as defined in Bytes 32 to 35 are listed in Table 2.3. The shapefile is restricted to this single type of shape, so other types must be defined in different shapefiles. The Bounding Box entries store the minimum confining rectangle, or cuboid, which encloses all shapes in the file. X and Y values usually refer to measures of longitude and latitude. Z values are most commonly used to represent elevations, but sometimes also rainfall or air quality. M values store linear measurements for route features, for example distances in meters between vertices of a pipeline.

Each record of the main file also contains its own header of 8 bytes to store record number and content length of 4 bytes each. Record numbers are ascending and start at 1. Shapefile record contents are composed of a shape type integer value (see Table 2.3), followed by the geometric information of the shape. The length of a record is variable and

Position	Field	Value	Type
Byte 0	File Code	9994	Integer
Byte 4	Unused	0	Integer
Byte 8	Unused	0	Integer
Byte 12	Unused	0	Integer
Byte 16	Unused	0	Integer
Byte 20	Unused	0	Integer
Byte 24	File Length	File Length	Integer
Byte 28	Version	1000	Integer
Byte 32	Shape Type	Shape Type	Integer
Byte 36	Bounding Box	Xmin	Double
Byte 44	Bounding Box	Ymin	Double
Byte 52	Bounding Box	Xmax	Double
Byte 60	Bounding Box	Ymax	Double
Byte 68	Bounding Box	Zmin	Double
Byte 76	Bounding Box	Zmax	Double
Byte 84	Bounding Box	Mmin	Double
Byte 92	Bounding Box	Mmax	Double

Table 2.2: Description of the Main File Header. Adapted from Esri Shapefile Technical Description [Esr98].

depends on the number of elements in a shape. The total size of the main file measured in 16-bit words is $H + (R * C_I)$ where H stands for the fifty 16-bit words of the header, R is the number of records, and C_I stands for the content length of record I.

The *index file* contains a 100-byte header that is identical to the aforementioned main file header. Following the header, there is a sequence of 8 byte records of fixed length. The total length of the index file is $H + 4 * R$. The *dBASE file* contains the feature attributes with a variable number of fields or columns for each shape. It follows the DBF standard, readable by many table-based applications such as LibreOffice [Esr98].

2.1.5 Visualization of Geographic Information

Spatial Autocorrelation and Scale

GIS is about representing geographic phenomena of the real world. The key to developing a well-founded representation of the real world is to build an understanding of the nature of spatial variation, proximity effects and level of detail (scale). In order to discuss the important property of geographic *scale*, the concept of *spatial autocorrelation* needs to be introduced. Spatial autocorrelation is dependent on spatial heterogeneity, which defines the degree of disparity of geographic places and regions. The disparity of places not only includes the appearance of the landscape, but also the observable processes that act on the landscape. Processes may oscillate about an average value, which is termed

Value	Shape Type	Description
0	Null Shape	Contains no geometric data.
1	Point	Consists of double-precision coordinates X, Y.
3	PolyLine	Ordered set of vertices that consists of $l - n$ parts, which are connected sequences of $2 - n$ points.
5	Polygon	Consists of $1 - n$ rings, connected sequences of $4 - n$ points that form a closed, non-self-intersecting loop.
8	MultiPoint	Represents $1 - n$ points.
11	PointZ	Consists of double-precision coordinates in the order X, Y, Z plus a measure M.
13	PolyLineZ	Ordered set of vertices that consists of $1 - n$ parts, which are connected sequences of $2 - n$ PointZs.
15	PolygonZ	A PolygonZ consists of $1 - n$ rings.
18	MultiPointZ	Features a set of PointZs.
21	PointM	Consists of double-precision coordinates X, Y, plus a measure M.
23	PolyLineM	Ordered set of vertices that consists of $1 - n$ parts, which are connected sequences of $2 - n$ PointMs.
25	PolygonM	Consists of $1 - n$ rings, connected sequences of $4 - n$ PointMs that form a closed, non-self-intersecting loop.
28	MultiPointM	Features $1 - n$ PointMs.
31	MultiPatch	A MultiPatch consists of a number of surface patches.

Table 2.3: Shape types values and description. Adapted from Esri Shapefile Technical Description [Esr98].

controlled variation. Longer-term processes like global warming oscillate in a growing range, termed *uncontrolled variation*. There are geographic phenomena that change smoothly across space, while others vary extremely abruptly, violating Tobler's Law. Taking autocorrelation into account is helpful in implementing geographic representations, but leads to complex and error-prone predictions [Lon+15].

Spatial autocorrelation measures the degree to which one object is similar to other neighbouring objects. In order to compute the similarity between neighbours, the most widely used statistics is the Moran Index [Mor50]. It is defined within the interval $[-1, +1]$, positive when nearby objects have similar attributes and negative when they have dissimilar attributes. The statistics returns zero, when attribute values are arranged randomly. This is the case when attributes are independent of location and space. The formula for calculating the Moran Index is defined by the following expression [Par09]:

$$I = \frac{n}{S_0} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2},$$

where n is the total number of features, w_{ij} is the spatial weight between i and j, and S_0 is the sum of all spatial weights w_{ij}:

$$S_0 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}.$$

Figure 2.6 shows three types of spatial autocorrelation and their corresponding values of the I statistic. In this particular case of autocorrelation the attribute data is, according to the taxonomy presented in Section 2.1.3, *nominal*. The field objects have two distinct colors with no implied order and a constant single value in every measured location within the field. Assuming measures at intervals that do not coincide with the length of the square tiles as shown in Figure 2.6, the resulting values for the spatial autocorrelation would be divergent. This means that the sampling interval is dependent on scale. Scale in the of context of GIS is therefore relevant for the compromise between attribute details and the level of geographic resolution. This trade off and its implications are relevant scientific questions. Choosing a sensible scale for creating a representation for a geographic application is a non-trivial task. Similarly, there is also the problem of selecting what to include or to omit in digital representations of the reality. In an effort to further restrict the meaning of the term scale, in this thesis the term is used as synonym to the level of detail or spatial resolution in data. In the context of maps, scale refers to the map's *representative fraction*, defined by the ratio of the distance on the map compared to the distance in the real world. A pattern that is closely related to spatial autocorrelation and scale is *self-similarity*. This property describes structures that replicate themselves at finer levels of detail. These patterns occur in nature as well as in social systems. Coastal features for example often repeat themselves in structure and form and municipalities tend to feature similar attributes [Lon+15].

Georeferencing

Chapter 2.1.3 introduced the tuple of geographic data: place, an optional time and an attribute component. The location data is not optional for GIS and this chapter focuses on providing techniques for assigning values on location. Location is the central part of the GIS and enables mapping, spatial analysis and operations. *Georeferencing* [Hil09] is the method of assigning a set of information to a specific location. The georeference must be unique, so there is only a single location mapped to its reference. Of course it is possible to link multiple items of information to a common location. Inherent to every georeference is its spatial resolution, which equals the area linked to that reference. A state of Austria has

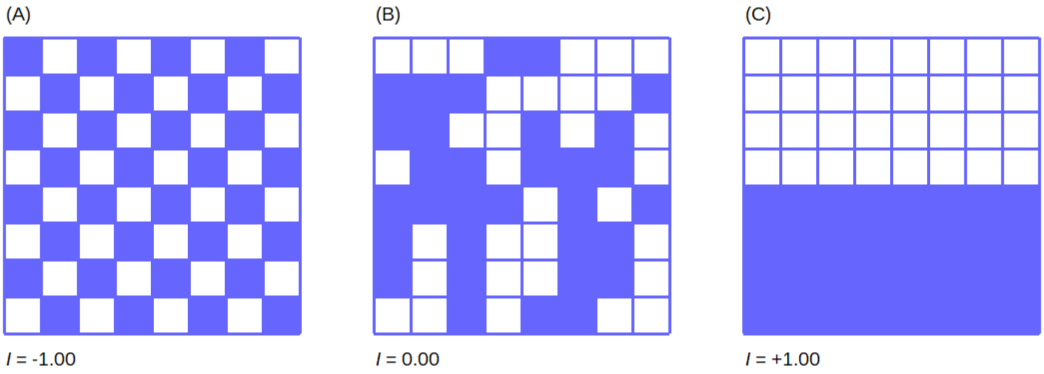


Figure 2.6: Example of three different instances of spatial autocorrelation using a $8x8$ space of blue and white squares displaying: (A) negative spatial autocorrelation; (B) a random arrangement; and (C) positive spatial autocorrelation.

a spatial resolution that varies from the area of Vienna (415 km^2) to that of Lower Austria (19 180 km^2) [STA21]. So in order to build a map representation, a referencing system must be established. Many georeferencing systems have been developed from simple, name-based and metric systems, to complex cadaster and coordinate-based systems. One relevant attribute to classify georeferencing systems is their spatial resolution. More advanced systems allow infinitely fine spatial resolutions, depending on the accuracy of measuring devices, and they support computation of distances - a key use case of GIS [Lon+15].

Nominal systems are the most simple form of geographic references and refer to giving names to geographic objects. Place names are widely used today, for example for referring to oceans, continents, countries, states, cities, mountains, rivers and many other prominent geographic features. Problems arise when multiple names are assigned to the same feature, like in different languages. Another issue is the alleged coarse spatial resolution. The information that a building is located in Vienna is of limited use as georeference, because there are thousands of other buildings in Vienna and some kind of additional information is missing. This problem is solved by postal addresses.

Postal addresses are established globally and generally georeference human activity such as defining the place of residence or business and enabling postal services. This system provides a unique identifier for every building or flat in the world, if the information in every layer is in turn unique within the parent-layer: Street number, street name, postal code encoding district and city, and finally country. While this system works for man-made entities, it fails for referencing natural features like mountains, rivers, or a specific spot in a forest. Postal addresses do not support spatial operations such as computation of distances. Only rudimentary calculations and inferences are possible assuming dwellings are numbered consecutively, such as estimating that the distance between two buildings with street numbers 4 and 28 is greater than the distance between

7 and 11.

Another approach is used by *linear referencing.* This system first defines a fixed point in space and then measures the distance along a known path in the network. This approach is widely used in infrastructure applications that rely on linear networks. This especially includes railroads, highways, pipelines and power supply lines. For example, highway agencies georeference construction sites or accident locations by defining the fixed point, the distance and the direction, for example *'highway A1, kilometer 120, towards Salzburg'*. In practice, linear referencing is often combined with postal addresses, like *'100 meters west of Main Street 20'*.

Historically, one of the first mapping and GIS applications was managing records of land ownership. Parcels of land in the so-called *cadaster* are usually identified by their number. One of the most prominent representative of cadaster systems is the *US Public Land Survey System* (PLSS), founded in the early 19th century. The PLSS is a system of surveying and dividing land in the United States, controlled by the Department of the Interior, Bureau of Land Management (BLM). Today, the system encompasses the majority of 30 states; Texas, Hawaii and 18 eastern states historically implemented other systems. The PLSS system is described in Figure 2.7. The land is divided into so-called *townships*, squares of six miles side length. The survey starts at a fixed point, defining it as the intersection of virtual north-south and east-west lines. The north-south line that runs through the fixed point is the *Principal Meridian.* In PLSS there are 37 Principal Meridians and a unique name is assigned to each of them. The east-west line is perpendicular to the Meridian and called *Base Line.* Townships are surveyed along those lines and subdivided into 36 sections of one mile squared. Sections are further subdivided into quarter-sections, half of quarter-sections and quarter of quarter-sections. In order to mark the areas, a permanent marker is placed at each subsection corner. Townships are referenced by the township designation and the range designation related to the center point, for example 'Township 2 South Range 3 West', or short 'T2S R3W' like depicted in Figure 2.7. A reference to a section includes state, name of the Principal Meridian, Township and the section number: Mississippi, Choctaw Meridian, T7N R2W, sec 14 [U.S12].

The georeferencing system that is widely used today, supporting arbitrarily fine spatial resolutions, is the *geographic system of coordinates*, or also called longitude and latitude. As a metric system, it also allows spatial analysis and the computation of distances between locations. Longitude and latitude are defined by the axis of the Earth's rotation, which runs from the North Pole to the South Pole and through the center of its mass as shown in Figure 2.8. The *Equator* is defined by assuming a plane through the center of the mass perpendicular to the axis. The lines of longitude, or Meridians, are defined by virtual lines connecting North and South Pole, perpendicular to the Equator. By convention, the Prime Meridian running through the Royal Observatory in Greenwich, London, is defined with zero longitude. In order to determine the longitude of a point on the surface of the Earth, the angle between a plane through the point and the axis, and the Prime Meridian is measured. Longitude is measured in degrees ranging from zero

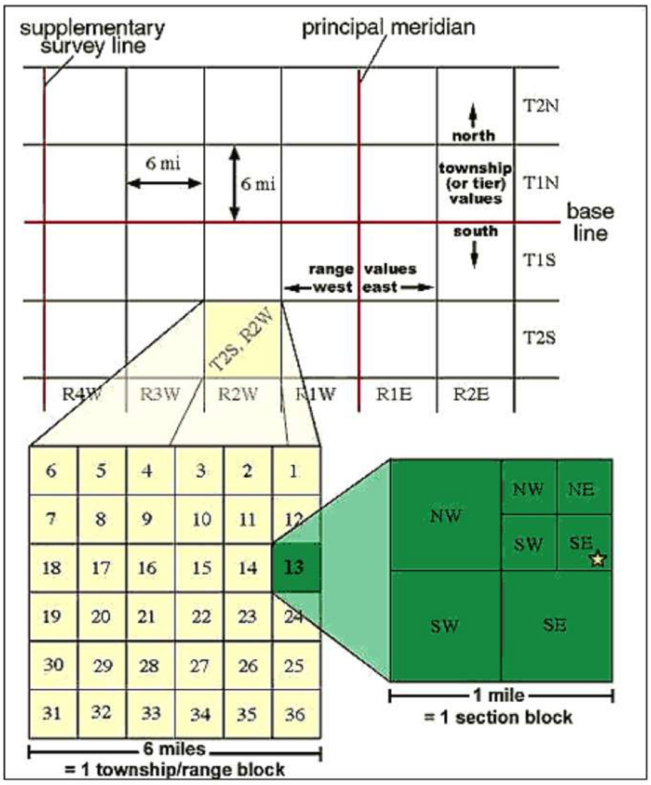


Figure 2.7: Basic PLSS township referencing and numbering [Ida08].

degrees at the Prime Meridian in Greenwich to 180 degrees East and 180 degrees West. Each degree is divided into 60 minutes and each minute is divided into 60 seconds. While this notation is more suitable for humans, computers process floating point numbers. As a result, fractures of degrees of longitude are stored in decimal number notation instead of minutes and degrees. For that method, West longitude is stored as a negative decimal number and East longitude is stored as a positive decimal number [Lon+15].

For the definition of latitude, the *figure of the Earth* is of important matter. The Earth is rotating about its shorter axis and as a result only approximately spherical, termed *spheroid*. The *flattening* is defined by the difference of the major (a) and the minor axis (b) relative the the major axis: $f = (a - b)/a$, which results in about one part in 300. Years of research had been invested in finding most realistic ellipsoids in order to produce accurate maps and requirements of international cooperations in the fields of military and aviation, as well as new sensing data from satellites accellerated the finding of an international standard. The World Geodetic System of 1984 (WGS84) by the U.S. Department of Defense [Nat00] is the widely accepted referencing system and also used by GPS. It defines the radius at the Equator a =6 378 137 m, the shorter polar radius b =6 356 752 m and the flattening $f = 1/298\,257$. In order to determine the latitude of a

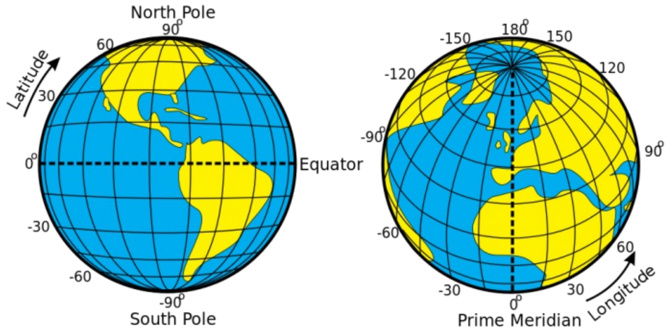


Figure 2.8: Latitude and longitude of the Earth [Com11].

point on the surface of the Earth, the angle between a virtual line drawn perpendicular to the ellipsoid of the Earth and the Equator is measured. Latitude ranges from 90 degrees South to 90 degrees North. Usually, longitude and latitude are represented by the Greek letters phi (ϕ) and lambda (λ). One degree of latitude between two points on the same Meridian corresponds to about 111 km, one minute equals 1.86 km, defining one nautical mile, and one second corresponds to 30 m. At the Equator, the lines of longitude are furthest apart and the distance diminishes towards the poles. The degree of the decrease is approximated by the cosine of latitude ($\cos\phi$). It is possible to calculate the distance between any given pair of coordinates. Assuming a spherical Earth, because the flattening would make the calculations much more complex, the shortest path between two points is along the *great circle*. This is the largest circle that can be drawn on the surface of the sphere and divides the sphere in two equal hemispheres. The formula for calculating the distance D on a spherical Earth of radius R between two coordinates is:

$$D = R \arccos[\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos(\lambda_1 - \lambda_2)]$$

where the subscripts refer to the two coordinates A(ϕ_1, λ_1) and B(ϕ_2, λ_2) [Lon+15].

For practical reasons, GIS have to deal with the flat surface of a *projected* Earth. One factor is that visualization of geographic data such as paper or screens are flat. Paper is still a relevant medium for digitizing or printing of geographic data for map production. Another reason is that raster data is by definition two-dimensional (see Chapter 2.1.3) and finally photographs captured by remote sensing of aircraft or satellites used in GIS are also mainly two-dimensional. Developments like terrestrial laser scanning (TLS) [Lem11] and light detection and ranging (Lidar) [Nat12] produce three-dimensional representations, which are still displayed on flat mediums. Therefore, projections are needed to handle the discrepancy between the earth ellipsoid and flat rasters. A projection is a function that transforms a position in Cartesian coordinates (x, y) from the raster into a position on the Earth's surface defined by latitude and longitude (ϕ, λ) or vice versa. The widely used Mercator projection is defined by the following pair of mathematical functions:

$$x = \lambda$$
$$y = \ln\tan(\phi/2 + \pi/4)$$

where ln refers to the natural logarithm function. Projections inherently distort the representation of the Earth, so it is not possible to preserve the scale, or relative distances between any two points. Two properties of projections are particularly relevant, although only one of them can be achieved. The conformal property ensures that angles between any two lines remain the same, so the shapes of small features are preserved. This is especially useful for navigation and topography, where straight lines keep a constant bearing. The equal area property guarantees that areas keep the same proportions, which is required for computation of surface areas. Another classification of projections is by analogy to a geometric model related to the position of the map compared to the Earth and three major classes are distinguished: *cylindrical*, *azimuthal* and *conic* projections. These three types are either of conformal or of equal area property. The Universal Transverse Mercator (UTM) projection is widely used in military and global datasets. The UTM system is cylindrical and conformal, so small features have the correct shape, but not size. The Earth is divided into 60 zones and each of them is six degrees wide. The coordinates of a UTM zone are metric, so accurate calculations of distances are possible. Coordinates are defined by a pair of a six-digit integer and a seven-digit integer, sometimes supplemented by a zone number, e.g. $32U4613445481745$ [Lon+15].

Cartography

Maps as output of GIS are the main medium of visualization of geographic information. By producing a visual representation from results of GIS operations, geographic information can be summarized and communicated effectively to a wide audience of both professionals and consumers. Most people exclusively interact with GIS by utilizing their map products. Longley et al. define the term cartography as the „art, science, and techniques of making maps or charts“ [Lon+15]. By convention, maps refer to terrestrial areas and charts refer to marine areas. Paper is still a relevant medium of transport of geographical information due to its several advantages. Paper maps are easy to transport, reliable, intuitively used and a basic application of printing technology. Yet, with increasing accuracy of our understanding of the world and with increasing complexity of the applications and computations involving geographic data, the demands for maps as transport media become more sophisticated. In the present, an enormous range of devices enables a worldwide community of users to benefit from a growing spectrum of geographic decision-making options that offer more features than paper maps. Navigation displays in vehicles, smartphones and wearable devices bring GIS applications to billions of users.

In the context of this thesis, a map is a digital or printed result of a GIS representing geographic information by complying with cartographic conventions. Maps that display two or more variables are called *multivariate maps*. In order to create a map, a GIS applies a series of processing methods to geographic data, transforming geographic information

Map Attribute	Paper Map	Digital Map
Scale	fixed	variable (zoom)
Extent	fixed	scrollable (pan)
View of the world	static	animated
3D data support	limited	interactive
Data layers	fixed	variable

Table 2.4: Comparison of paper maps and digital maps. Adapted from [Lon+15].

to visual data. Important steps of GIS-supported map creation are data collection, editing, management, analysis and output. The literature differentiates between reference maps and thematic maps. Reference maps, such as topographic maps, show geographic features in relation to each other and thematic maps, like presentations of world time zones, show a particular attribute connected with a spatial area. Maps are a means to communicate information, present results of analysis, display spatial relationships and can assist in decision support. One limitation is that maps are a single instance of a geographic process. Each map represents one sample of all possible result sets. As a consequence, other map samples generated from the same environment would result in variations. For example, sampling of soil textures will show natural variations [Lon+15].

GIS fundamentally changed analog cartography and map creation. Several limitations are inherent to paper maps, while digital maps are a more feature-rich and interactive medium. See Table 2.4 for a comparison of paper and digital maps.

Main principles of the map design process were formulated by Robinson et al [Rob+95] as seven controls. The *purpose* of a map defines the contents and how the geographic data is presented. The *reality* forces constraints such as orientation of a country. The specific attributes of available map *data* affect symbolization. The concept of *scale* influences the amount of data displayed and the size of symbols and other map objects. Different *audiences* require different types and shaping of maps. Advanced users are able to process complex information, while basic users or children prefer summary information. The *conditions of use* require specific designs, for example to facilitate usage in poor light or under water. *Technical limits* of the display medium influence the design process of maps. For example, it should be taken into account that the screen size of handheld devices is much smaller than the screen size of desktop monitors.

Map composition is another aspect of map production and defines the different components of a map as illustrated in Figure 2.9. The principal element of a map is its main *map body*, the actual geographic map. Comparative maps feature two or more map bodies. *Inset maps* are often used to display an area of the map body in greater detail, or the context of the map body. The *legend* lists the map objects and their symbolization. The legend is essential for reading and understanding the map. The *title* identifies the contents of the map and the *map scale* defines the ratio between one unit on the map and in the real world. A coarse map scale represents a larger area on the map, but contains

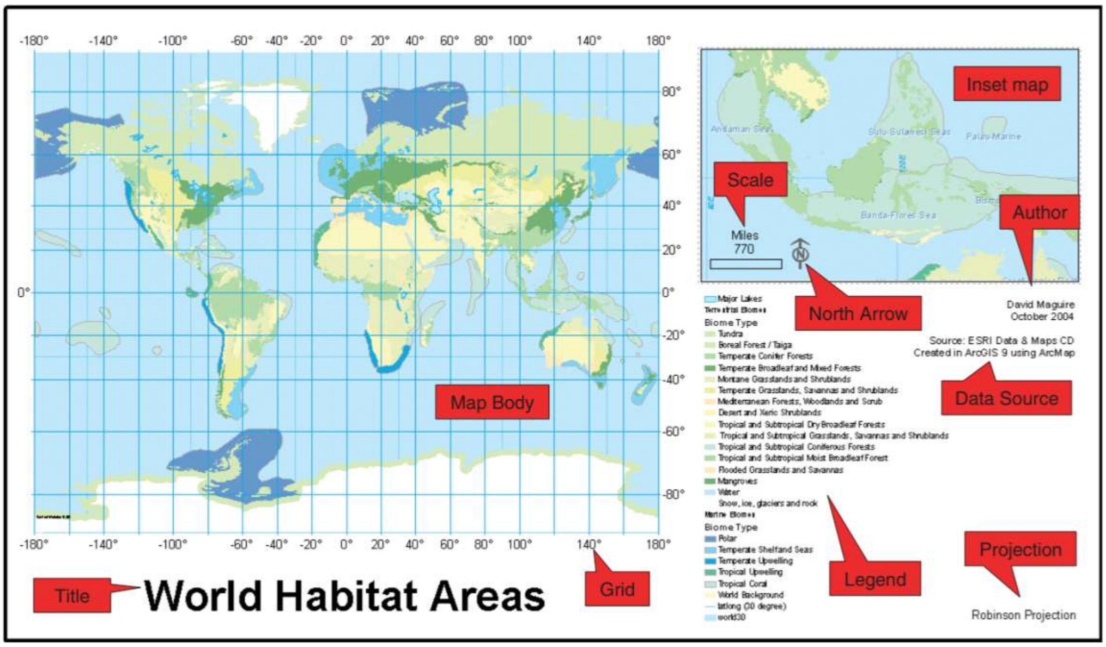

Inset map
Scale
Miles
770
Author
North Arrow
David Maguire
October 2004
Source: ESRI Data & Maps CD
Created in ArcGIS 9 using ArcMap
Data Source
Map Body
Major Lakes
Terrestrial Biomes
Biome Type
Tundra
Boreal Forest / Taiga
Temperate Conifer Forests
Temperate Broadleaf and Mixed Forests
Montane Grasslands and Shrublands
Temperate Grasslands, Savannas and Shrublands
Mediterranean Forests, Woodlands and Scrub
Desert and Xeric Shrublands
Tropical and Subtropical Dry Broadleaf Forests
Tropical and Subtropical Grasslands, Savannas and Shrublands
Tropical and Subtropical Coniferous Forests
Tropical and Subtropical Moist Broadleaf Forest
Flooded Grasslands and Savannas
Mangroves
Water
Snow, ice, glaciers and rock
Marine Biomes
Biome Type
Polar
Temperate Shelfand Seas
Temperate Upwelling
Tropical Upwelling
Tropical Coral
World Background
latlong (30 degree)
world30
Projection
Legend
Robinson Projection
Title
World Habitat Areas
Grid

Figure 2.9: The components of a map composition [Lon+15].

less details. The *orientation* of a map can be represented by a north arrow, a grid, or a graticule. A grid is composed of parallel and perpendicular lines and a graticule is a network of latitude and longitude lines. *Map metadata* contains additional information such as author, data source, projection, or creation date [Lon+15].

Map symbolization is the process of classification of geographic objects, as introduced in Chapter 2.1.3, using conventional graphic symbols. Depending on the map scale and relevance to the application, spatial objects are represented as points, lines, areas, or mixtures of them to abstract reality. Attributes are similarly mapped as symbols, points, lines, areas and color [Rob67] in order to display different types of information. Bertin's investigations regarding the significance of symbols [Ber83] were later refined by MacEachren [Mac95], who developed a typology of ten graphic primitives in cartography. In order to present only the most important concepts, *size* and *orientation* of symbols is varied to represent ordinal and interval data. The use of color is described by *color hue*, *value* and *saturation*. Color hue is defined by the dominant wavelength, the color value is defined by light of dark variations of a color hue and finally the color saturation is measured by the intensity of a color hue. Colors usually describe *nominal attribute* categories, like forests, sea and urban areas. Colors may be combined with *shapes* and *textures* in case of a large number of attribute categories. *Arrangement* of symbols convey directions and patterns like ocean currents [Lon+15].

A common task for GIS is the optimal positioning of nominal data representations on the map in order to enhance map interpretability. GIS applications usually include algorithms for placing labels and symbols in relation to their corresponding geographic object while avoiding overlaps. Ordinal attribute data is represented similarly by the application of a hierarchy of graphical symbols through sizes, types, colors, or intensities. The differentiation of interval- and ratio-scale attribute data is achieved by a range of conventions depending on the spatial object type. Point locations are often mapped by proportional circles, lines are distinguished by width or color, and areas are depicted by zones. Such maps, constructed from non-overlaping areas, are called choropleth maps. Two issues are inherent to choropleth maps, which have the potential to transport misleading messages. First, the attribute value is uniform within each zone. Second, large but less relevant areas may dominate the map visually. As a result, four classification schemes have been developed to normalize interval and ratio data. *Natural (Jenks) breaks* classify the underlying data by natural groupings, which are relevant to the specific application. The second scheme is *quantile breaks*, where a certain number of classes contains an equal amount of observations. Quartile, or four-category, classifications are often applied in statistics. Quintile, or five category, classifications are commonly used for displaying uniformly distributed data. Quantile breaks may cause visual distortion when almost identical attributes are assigned to adjacent classes, while very different attributes end up in the same class. *Equal interval* breaks are applied when the data ranges are well-known to the audience, like a temperature scale in degree Celsius. The fourth scheme to be discussed is the *standard deviation* classification. First, the mean value is computed and then class breaks in standard deviation are calculated above and below the mean [Lon+15].

2.1.6 GIS Software

In Chapter 2.1.1, the components of a geographic information system were defined as hardware, software, data and people. Geographic data models have been introduced in Chapter 2.1.3 and the output of a GIS, the visualization of spatial data, has been discussed in Chapter 2.1.5. This chapter is dedicated to GIS software, the GIS component with the capability to operate on data in order to produce a useful result for people. Software products with limited, single-purpose capabilities like simple graphics and map display, location-based services, navigation or image-processing applications do not qualify as GIS.

Architecture

The architecture of a typical GIS corresponds to the standard three-tier architecture of information systems: presentation layer, business layer and data access layer. Accordingly, GIS are built of three key parts: user interface, tools and data management. The graphical user interface (GUI) enables the user's interaction with the system. The GUI consists of control elements like menus, bars, buttons and windows. These control elements expose GIS tools, the second tier, to the user. The methods and algorithms of the GIS package for processing geographic data are defined by this toolset. The third tier, the geographic data, is stored in files or databases, which are administered by database management systems (DBMS). The demands on the three tiers are very different. The presentation layer is required to render raster and vector graphics and objects. The toolset, or business logic tier, must be able to perform compute-intensive algorithms such as overlay processing, raster analysis or routing. The data layer is responsible for input and output of geographic data from a file or a database system. In order to enhance overall GIS performance, the three layers are deployed on distributed, specialized hardware and operating systems. Depending on the application scenario, four variants of computer system architectures are commonly used to operate GIS software. When all three architectural tiers are installed on a single-user (desktop) computer, it is called *desktop* configuration. In a more complex multi-user environment the three presented layers can be deployed on distributed machines to improve overall performance and flexibility. This configuration is referred to as *client-server*. For example, the desktop GIS installed on a desktop computer contains the GUI and the business logic layers, but the geographic data and the DBMS reside on a server connected over a local area network (LAN), wide area network (WAN), or over the Internet. In an even more distributed configuration, the *centralized desktop architecture* moves GUI and business logic on a centralized server. In contrast to the thick client of the client-server architecture, the GIS package is installed on an application server. Users have remote access to the software via terminal PCs, or thin clients. Additionally, a dedicated data server is deployed for improved performance. This architecture enables high-end capabilities such as spatial analysis, simulation and advanced editing for large organizations. In a variant of the centralized desktop architecture, the business logic is deployed on a dedicated server, where a range of desktop PCs, thin clients, browser clients and embedded clients communicate with the middle tier server. This *centralized server GIS* is used in large and complex departmental and enterprise implementations [Lon+15].

In the present, professionals commonly work with GIS software that runs on desktops or on distributed systems. One reason for the paramount status of desktop GIS is interoperability with other desktop applications like spreadsheets, databases or rendering tools. Other advantages of desktops are the relatively low costs and high performance with multi-core processors. Common GIS desktop applications are mapping, geographic data editing, 3-D visualization, spatial modeling and analysis. Typical organizations for desktop GIS are small and medium-sized businesses (SMBs), educational- and governmental facilities. Even though desktop GIS products remain successful, network-based GIS have the potential to lower the cost of ownership and enhance access to geographic information. One reason is that network-based GIS are implemented on centralized business and data layers, which can reduce deployment, support and maintenance costs. Such distributed GIS can use a cross-platform Web browser software to host the GUI layer. The client itself only features visual output and simple data query capabilities, while the business and data tiers are deployed on different server systems, corresponding to the previously presented *centralized server GIS*. Recent developments aimed to combine the advantages of the desktop and network architectures towards a *rich client* architecture. In this approach, the application is downloaded dynamically from a server on user request [Lon+15].

In the context of GIS architecture, two more concepts are relevant. The first concept, *data models*, has been discussed in Chapter 2.1.3 with a focus on raster and vector data. The data model determines how the reality is represented in a GIS. *Customization* is the second concept to be presented. It describes the process of configuring GIS software, developing additional functionality, or embedding modules in other systems [Lon+15]. The customization process ranges from simple GUI modifications to sophisticated third-party extensions providing new features. In order to be able to communicate with other applications, the GIS must expose certain functions by interfaces. These application programming interfaces (APIs) allow third-party tools to call functions of the GIS software. For the simulation part of this thesis, the simulation software Agent Analyst communicates with the API of the GIS software ArcGIS for visualization, spatial computations and data storage. Please refer to Chapter 5 for more details about the implemented simulation.

State-of-the-art GIS software systems have three main features in common. The first basic functionality is a data management system for geographic data. Another essential feature is a visualization package for displaying maps and other geographic output. The third component is a system for spatial modeling and analysis for manipulating spatial data using toolkits. In the following section there is a discussion on selected commercial and open-source desktop and server GIS software packages that fulfill these three requirements.

Software Types and Products

GIS software, as defined at the beginning of this chapter, can be classified in four main types: desktop, server, developer and mobile GIS. Software may additionally be differentiated by its business model and distinguished in commercial, or proprietary software, and open-source software. The most widely used category of GIS software is desktop GIS. A comprehensive list of commercial desktop GIS software products is provided in Table 2.5. This type includes a wide range of packages, starting from basic, mostly free of charge, mapping tools like Esri ArcGIS for Desktop Basic, Precisely (formerly Pitney Bowes) MapInfo ProViewer and Hexagon GeoMedia Viewer. Commercial desktop GIS software products with an advanced feature set are Autodesk Autocad Map 3D, Hexagon GeoMedia Essentials and Advantage, Precisely MapInfo Professional and Esri ArcGIS for Desktop Standard, which is used for the simulation part of this thesis (see Chapter 5.1 Integration of GIS and simulation software). High-end products deliver features like data manipulation, advanced capabilities for editing, and geostatistical and topological analysis to professional users. This category includes products such as Hexagon GeoMedia Professional, General Electric Smallworld GIS and Esri ArcGIS for Desktop Advanced. Desktop GIS users commonly work in areas like engineering, administration, teaching, business planning, marketing and similar professions [Lon+15].

Server GIS products are gaining relevance due to the potential for more concurrent users and lower cost per user compared to classic desktop GIS packages. This development is a result of the shift towards mobile devices with a constant connection to the Internet [Aus15] and the market demand for ubiquitous geographic computing. Software vendors

Software Package	**Vendor**	**Type(s)**	**Features**	**Support**
ArcGIS	Esri, Inc.	All	Offers a full range of 2-D and 3-D GIS products depending on need, including simple visualization and modeling to advanced analysis and presentation.	One-year standard support included with software purchase. Premium support with 24/7/356 telephone and internet help available for additional fee. [ESR17a]
Bentley Map	Bentley Systems, Inc.	Desktop, server	Primarily concerned with infrastructure; full range 2D and 3D mapping application with additional modeling and presentation support in extended editions.	24/7/365 available with *Bentley SELECT* license agreement. [Ben17]
Autocad Map 3D	Autodesk, Inc.	Desktop	Model-based software with a focus on infrastructure. Provides comprehensive access to GIS data.	Online forums and documentation available. Autodesk subscription available for purchase with access to latest releases and expedited technical support. [Aut17]
GeoMedia	Hexagon AB	Desktop, web mapping, server	Full range software suite that includes data intake and management, mapping, modeling, presentation, and distribution.	Included with license.
Mapinfo	Precisely Software	Desktop	Mapinfo is a mapping tool for capture, editing, analyzation, visualization and presentation of geographical data. Supports MIF/MID interfaces.	Free tutorials, knowledge base and online forum. License includes a 1-year maintenance support. [Pre21]
Disy Cadenza	disy Informationssysteme GmbH	Desktop, web, mobile	Cadenza is a platform for comprehensive reporting and analysis of factual and spatial data. GISterm enables all kinds of users to create, process, manage, analyze and visualize spatial data.	Licenses and service terms upon contact. [Dis17]

Table 2.5: A selection of commercial GIS products for desktop systems.

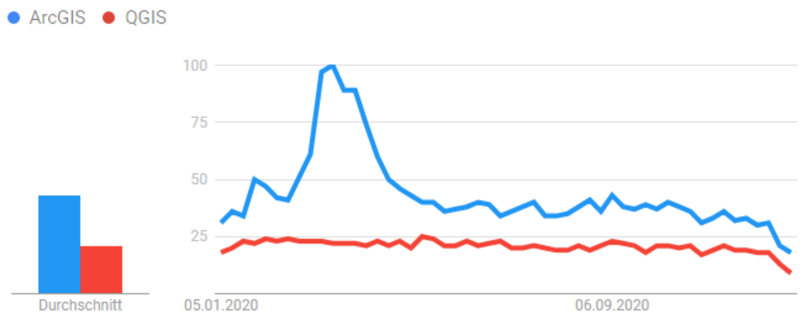


Figure 2.10: Comparison of search-engine interest in ArcGIS versus QGIS for the year 2020 (source: Google Trends). The averages are 43 (ArcGIS) and 21 (QGIS).

adapted this trend by releasing server-based packages such as Esri ArcGIS Enterprise, Autodesk Infrastructure Map Server, General Electric Smallworld Internet Application Server, Precisely MapXtreme and Hexagon GeoMedia Webmap. Server GIS must be able to handle many concurrent requests from remote clients like mobile apps and Web browsers. Server GIS offer spatial features and toolboxes similar to desktop GIS packages.

Developer GIS are products targeted for software developers. They do not work standalone, but are used to build customized applications in embedded systems. Typical developer GIS modules contain spatial visualization and data query capabilities. Representatives of this GIS software branch include Precisely MapInfo MapX, Blue Marble Geographics GeoObjects and Esri ArcGIS Engine. Most of these components are built on the Microsoft .NET and on the Oracle Java platform. The area of application is not limited, but prominent deployments are in navigation systems, customer care systems or resource planning [Lon+15].

The hardware of mobile devices such as smartphones, tablets, portable navigation systems, and smartwatches improved and high-end devices even offer capabilities comparable to desktop systems. The prominent features of mobile devices are wireless networking and location positioning support, which enable mobile applications to communicate with a server GIS in order to display and query geographic information.

Beside commercial GIS products, there is a growing range of open-source software as listed in Table 2.6. In terms of features and software quality, most of these packages are comparable to commercial products, especially the successful application suite QGIS. For a comparison of commercial and open-source desktop GIS there are no market share numbers available, but the Google search trends of ArcGIS and QGIS[11] compared in Table 2.6 proves the relevance of open-source GIS. The next paragraphs present the two predominant open-source GIS packages QGIS and GRASS GIS.

QGIS is licensed under the GNU General Public License and it is an official project of

[11] `https://trends.google.com/trends/explore?q=ArcGIS,QGIS` (visited on 03/19/2021)

the Open Source Geospatial Foundation (OSGeo). QGIS supports multiple operating systems, including Linux, UNIX, Mac OSX, Windows and Android. The QGIS project covers different use cases by providing applications for desktop, server, browser and client systems. The major features of QGIS are visualization of raster and vector data, manipulation of spatial data, spatial analysis and extensible plug-in architecture. The features of QGIS are accessed by a GUI, which has been translated into more than 40 languages. QGIS offers web services according to OSGeo interface standards[12] to allow processing of spatial data from external sources. To further extend its functionality, QGIS supports integration with third party Open Source GIS programs, for example GRASS GIS and MapServer[13]. QGIS supports plug-ins written in C++ or Python [QGI17] [Jam08].

Geographic Resources Analysis Support System (GRASS GIS) was originally developed by the U.S. Army Construction Engineering Research Laboratories (US-CERL). The open-source GIS is released under the GNU General Public License (GPL) since 1999. Since 2006, the Open Source Geospatial Foundation (OSGeo) supports the development of GRASS GIS as one of its founding projects. GRASS GIS is available on multiple platforms, including Microsoft Windows, Mac OS X, FreeBSD and Linux. It provides both a GUI as well as a shell for executing GRASS commands. Additionally, other GIS applications such as QGIS can directly interact with GRASS modules via command shell. GRASS GIS capabilities include raster, 3D-raster, vector and point data analysis, image processing, visualization and map creation. GRASS supports several databases such as DBF, SQLite, PostgreSQL, MySQL, ODBC, and provides interfaces to statistical software like R or Matlab. In order to provide these features, GRASS currently consists of over 350 modules and tools [GRA15] [Net00].

Esri ArcGIS

Esri, Inc. (former Environmental Systems Research Institute) is the market leading company in the GIS industry [ESR15]. Esri was founded in 1969 by Laura and Jack Dangermond and it is situated in Redlands, California, USA. In 2016, the company is still held privately, employs over 3200 people in the United States alone and yields revenues of 1.1 Billion US-Dollar [Hel16]. According to Esri, more than 350 000 customers including enterprises, governments, NGOs and academic organizations are using Esri's product portfolio, which is centered around the ArcGIS family. The market share of ArcGIS products is over 40 % of the GIS software market worldwide [Mih17]. The ArcGIS family consists of software for desktop, server, developer, mobile and online GIS [ESR17a].

The desktop GIS software *ArcGIS for Desktop* is distributed in three different versions depending on the application area. While ArcGIS Basic only contains elemental features to view and edit GIS data, ArcGIS Standard provides more functionality in the areas

[12] https://live.osgeo.org/archive/8.0/en/standards/standards.html (visited on 03/19/2021)

[13] http://mapserver.org (visited on 03/19/2021) MapServer is an Open Source platform for publishing spatial data and interactive mapping applications to the web.

Software Package	**Developer**	**Type(s)**	**Features**	**Support**
Quantum GIS	QGIS development team	Desktop, web mapping	Data management, mapping, integrates with other open-source software for extensive functionality.	Online documentation. Third-party support available for free. [QGI17]
MapWindow GIS	MapWindow Open Source Team	Desktop	Includes data viewer, modeling system, image processing and map-making tools.	Online community with forums and documentation available. Third-party support available for a fee. [Map17]
GRASS GIS	GRASS development team	Desktop	Includes data viewer, modeling system, image processing and map-making tools.	Free online book available. Online community with forums and documentation available. [GRA15]
Opticks	Ball Corp.	Desktop	Opticks is an expandable remote sensing and imagery analysis software platform that is free and open-source	Online forums, tutorials, mailing lists and chat available. [Bal14]
gvSIG	gvSIG association	Desktop, mobile	gvSIG is designed for capturing, storing, handling, analyzing and deploying any kind of referenced geographic.	Rich community with blogs, mailing lists, events. Documentation available. [gvS16]
openjump GIS	openjump GIS	Desktop	GIS data (GML files) viewer written in java.	User forums, mailing list and wiki available. [JVR17]
DIVA GIS	Robert Hijmans	Desktop	Mapping and geographic data analysis software.	Documentation, training manuals, tutorials and exercises available. [Hij11]

Table 2.6: A selection of open-source GIS products.

map creation and geodatabases. The high-end version ArcGIS Advanced includes 3D Analyst and other tools and features, especially for spatial analysis, and geoprocessing. Depending on the license version, ArcGIS is composed of up to four main components. *ArcCatalog* is a module for file and geodatabase management and it is included in all versions. It allows users to preview and manipulate spatial metadata. *ArcMap* is the mapping tool of ArcGIS and provides functions to display, edit and query geographic data. ArcScene is a component of the 3D Analyst package and enables visualization and analysis of 3D data. ArcGlobe is another application of 3D Analyst for viewing three-dimensional objects on a globe surface. The ArcToolbox provides geoprocessing functionality like computing distances, clipping, overlay and data conversion [ESR17b]. Esri markets a variety of extensions for ArcGIS for Desktop, for example analysis tools for business, networks and geostatistics, functions for maritime and aviation charting, and a pipeline, road and railway referencing system. It is possible to use extensions developed by third parties, built on APIs provided by ArcGIS for COM[14], .NET, Java and C++ platforms.

In the simulation part of this research, ArcGIS for Desktop is used in conjunction with Agent Analyst[15], a third party extension for ArcGIS. In order to prepare and transform raw geographic data from OpenStreetMap, several functions of ArcToolbox are required and the simulation output is visualized in ArcMap. For more details concerning the application and integration of ArcGIS for the implemented simulation please refer to Chapter 5.

2.1.7 Summary

This chapter provided an introduction to geographic information systems. Relevant definitions of GIS have been presented and the three dimensions of GIS have been discussed. The scale dimension represents the abstraction of the real world; the purpose defines the applied methods and the temporal aspect describes change over time. To understand the development of GIS, the history from the beginnings in the 1960s to the present, has been explored. A modern GIS is composed of software, data, procedures, hardware and users. The possible applications of GIS are multifarious. Five key-areas of GIS applications have been presented. In scientific context, GIS are used for problem analysis in the spatial domain. Governmental applications use GIS for example in natural resource management, infrastructure planning, transportation services, law enforcement and administration. The industry applies GIS products in production, marketing, transportation, logistics and many other areas.

A classification of geographic data has been provided. Geographic information can be viewed as discrete objects or continuous fields. The raster data model represents the world as rows and columns of square cells. In vector representation, all entities are made

[14] *Component Object Model*. Microsoft Corporation: `https://msdn.microsoft.com/en-us/library/ms680573` (visited on 03/19/2021)

[15] Agent Analyst: Agent-Based Modeling in ArcGIS. Esri, Inc: `https://resources.arcgis.com/en/help/agent-analyst/` (visited on 03/19/2021)

of points connected by straight lines. For visualization of geographic information, the concepts of spatial autocorrelation and scale are important. Systems for georeferencing, the method of assigning information to a specific location, have been introduced and discussed. Maps as output of GIS processes are the main medium of visualization of geographic information.

GIS Software is a collection of computer methods that operate on geographic data in order to produce a result. Common GIS architectures range from desktop systems to complex server infrastructures with different clients connected by a network. Software products have been classified in desktop, server, developer and mobile GIS. The most relevant commercial and open-source GIS products have been presented. Esri ArcGIS for Desktop is the world market leading GIS software. The integration of GIS and simulation is an integral part of the research documented in this thesis. With a basic understanding of geocomputing, the next chapter conveys the basics of computer simulation as the foundation for the presented geospatial agent-based simulation model.

2.2 Agent-Based Social Simulation

This chapter gives an introduction to agent-based social simulation (ABSS), a scientific method used in the context of this thesis. The focus lies in a discussion of the meanings and implications of computer simulation in social sciences using a multi-agent simulation approach. The study subject of the present thesis is social behavior of humans within a spatio-temporal dimension. Computer simulation is used in the field of social sciences since the 1990s as a suitable method of modeling and understanding social phenomena. Adding an agent-based approach to social simulation enables a new point of view on social processes based on the fundamental concept of emergence (see Chapter 2.2.6), where complex behavior emerges from comparably simple rules.

2.2.1 Introduction

In order to introduce the term simulation, it is necessary to discuss the concept of a model, because it is a fundamental part of a simulation. Similar to the meaning of the term *model* discussed in Chapter 2.1.1, a model in the context of simulation is a simplification of the real world. It is a smaller, less detailed, less complex, or all of these characteristics combined, image of a structure or a system. The phenomenon in the real world to be investigated is termed the *target*. The goal of the research is to design a model that is sufficiently similar to the target, so conclusions and findings about the model are also valid for the target [GT05].

Simulation is a form of building that model. The given literature does not provide a single, universally valid definition of the term *simulation*. A selection of suitable characterizations are presented in this chapter. The Oxford English Dictionary specifies simulation as „imitation of a situation or process“ [Oxf17]. The Encyclopædia Britannica provides a more comprehensive definition of the term, describing computer simulation as

„the use of a computer to represent the dynamic responses of one system by the behavior of another system modeled after it“ [The17]. They further state that the underlying model, which represents the real world, is in the form of executable code and when it is run, „the resulting mathematical dynamics form an analog of the behavior of the real system (...)“ [The17]. Finally, the goal of this method is „to study the dynamic behavior of objects or systems in response to conditions that cannot be easily or safely applied in real life“ [The17].

In an effort to combine different views in the literature, computer simulation can be described in a narrow sense as a program that is executed on a computer, using procedures and algorithms to examine the behavior of a specified model. The simulation program transforms input data to output data. That is, the procedure accepts as input variables a specific system state at a time t. That input is used to incrementally calculate the simulation state at time $t + 1$ until an end condition is met. The procedure creates a collection of output data, which can be viewed on a screen by means of visualization, or saved to a database or file system for further analysis. The step-by-step execution of the program is necessary when the examined model contains equations, for example continuous differential equations that cannot be solved analytically. According to Paul Humphreys, who developed a survey of main reasons why computer simulations are valid scientific methods, a computational approach is „any computer-implemented method for exploring the properties of mathematical models where analytic methods are not available“ [Hum90]. Still, computer simulations are also used either when the mathematical model contains discrete equations, or when when the mathematical model consists of a rule-based framework. Both types of models are suitable to be directly implemented in an algorithm and run as simulation. Discrete and continuous equation-based simulations are presented in Chapter 2.2.3 in greater detail. Discretized simulations only approximate the solution of continuous models at a certain degree of accuracy [Win19]. Rule-based models are the foundation of agent-based simulation (see Chapters 2.2.4 to 2.2.8) and do not depend on a series of differential equations. This type of computer simulation is one main scientific method of this study.

A more general characterization is suggested by Eric Winsberg in *Simulated Experiments: Methodology for a Virtual World*, where he postulates that simulation is rather a process than a single task: „Successful simulation studies do more than compute numbers. They make use of a variety of techniques to draw inferences from these numbers. (...) Much effort and expertise goes into deciding which simulation results are reliable and which are not“ [Win03]. In this sense, computer simulation is a scientific process for studying systems, which may be part of the real world or hypothetical systems. This process consists of formulating a suitable model, implementing the model in a programming language, executing the program on a computer, generating output data and finally studying the results in order to conclude inferences about the modeled target. Each of these fundamental steps are discussed throughout this chapter.

Applications of Computer Simulation

A study of the literature reveals several main purposes of computer simulation. In science, and especially social studies, simulations are applied to gain a better understanding of the world, for discovery and for prediction [Axe97]. But there are also usages of computer simulation in education, industry and practical contexts. The following paragraphs discuss a selection of relevant applications of computer simulation in various fields.

In social sciences, the major reasons for applying computer simulation as scientific method are in discovery and formalization. In social research, simple simulation models representing a small fraction of the world are built and their consequences are studied. If there is data describing how a system behaves, computer simulation can be used to investigate how and why these events actually did occur. The goal is to evaluate proposed theories in the simulated society or system. Formalization is the process of transforming theories defined in textual form into a specification that can be expressed as a computer programming language. In contrast to natural sciences like physics, social sciences did not incorporate mathematics as major tool for formalization, because simulation is a superior method for various reasons. First, computer programs are better suited to model reality than a set of equations. They support parallel processes, or concurrency, the execution of an algorithm in order or out-of-order, without affecting the final results. Parallel execution of program code can significantly increase the overall speed and efficiency of a simulation. Programs, especially those written in object-oriented programming languages, are modular, which enhances code reuse, extensibility and lowers maintenance effort. Finally, computer programs are able to simulate heterogeneous agents, which is discussed in greater detail in the following chapters focusing agent-based simulation [GT05].

Predictions, or forecasts, rely on a model that reproduces the functionality of a system. The goal is to learn how a system in the real world performs under a specific set of circumstances. This method can be used to make general or precise predictions, or also to retrodict the past. In order to increase the precision of a prediction, the simulation model needs to be more detailled and closer to the target system. Derived from the relative quality of the prediction, a proposed taxonomy distinguishes point predictions, qualitative predictions and range predictions [Win19]. In order to conduct a prediction, the simulation computes the system's state at a time $t + 1$ and thus mimics the passing of time. A classic application of this method is in demographic research where long-term developments of the age and size of a population is of interest. The corresponding simulation model would include observed fertility and mortality rates in order to predict changes in the future [GT05].

Another common use for simulation is to emulate decision-making of humans, so-called *expert systems*. Their goal is to simulate the capabilities of professionals like geologists or chemists, so non-experts are able to carry out analysis that would otherwise require expert knowledge [HRWL84]. A similar application of simulation is in training and education. For example, flight simulators are used to train pilots to reduce costs and risks. Urban simulation models like UrbanSim as shown in Figure 2.11 are used for

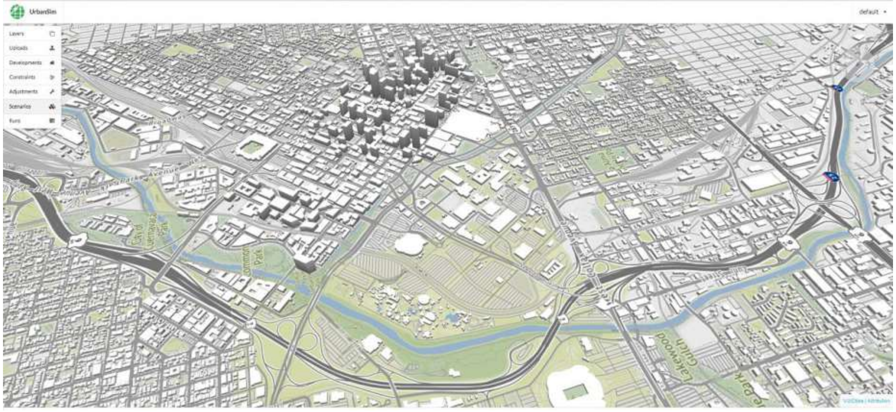


Figure 2.11: The UrbanSim Cloud Platform (UCP) is a web-based interface built to support planning and analysis of urban development, land use, transportation, the economy, and the environment. Source: UrbanSim [Pau17].

decision-support in urban development and related fields.

Simulations are also used to represent and to communicate knowledge, especially in education. It is advantageous to apply computer simulation in a classroom setting to enable students to discover natural phenomena and processes, such as bacterial studies, movement of planets, tectonic shifting or chemical reactions, which would be unfeasable or costly to observe in a laboratory or in the real world [Win19].

Simulations are also a major genre of the entertainment industry. Flight simulations, race simulations, business simulations, social simulations and other sub-genres have been developed to entertain and sometimes educate people with varying complexity of the simulated matter. One notable and influential representative is SimCity, a PC game, in which the player is developing and managing a city as a mayor (see in-game screenshot in Figure 2.12).

Historical Perspective

Several computer simulation methods are presented in their historical context. Chapter 2.2.3 discusses and characterizes the different simulation types in greater detail. Computer simulations evolved parallel to the development of computers. The first large-scale computer simulation was devised during the Manhatten Project. The simulation used a Monte Carlo approach developed by Stan Ulam and John Von Neumann to investigate the movement of neutrons through different materials [And+03] [Ato14].

In social sciences, the first applications of computer simulation occured in university research in the early 1960s. One method was a discrete event simulation, which modeled queues in order to simulate throughput, such as in traffic planning. Another early simulation approach, system dynamics, was based on a large set of difference equations [Ste00]. The Club of Rome studies of the future challenges of the world included the

Figure 2.12: SimCity (2013) is a city-building and urban planning simulation game developed by Maxis. Source: Electronic Arts Inc. [Ele14].

works of MIT Professor Jay Forrester, who created a system dynamics model of the world's socioeconomic system called WORLD1 and later refined as WORLD2 [For71].

Another simulation technique was developed in 1969 by the United States President's Commission on Income Maintenance Program. This microsimulation model called *Reforms in Income Maintenance* (RIM) was used to evaluate the consequences of governmental transfer programs depending on income as an alternative to existing welfare programs. The RIM model was extended in the 1970s and introduced as Transfer Income Model (TRIM), which was in turn superceded in 1980 by TRIM2. Since 1997, TRIM version 3 is developed and maintained at the Urban Institute[16] to simulate governmental tax, welfare, and health insurance programs in the United States [Urb12].

Microsimulation is based on a large set of units, representing individuals, groups of people, or organizations with certain attributes and states. During the simulation, these states are changed using transition probabilities to calculate estimates of the properties of the population in the future. This method is limited regarding the ability to explain phenomena; it is a tool for predictions only. Additionally, microsimulations are not able to model interactions between units. Despite the limitations, microsimulation was the dominant form of simulation in social sciences until the 1990s [GT05] [Har91].

The development of multi-agent, or also called agent-based simulation (ABS), changed research in social sciences radically. The new technique offered the simulation of individuals including their interactions for the first time. The origins of ABS derive from the research fields of nonlinear dynamics and artificial intelligence, where physicists and mathematicians developed cellular automata to simulate interactions between molecules

[16] Urban Institute, with primary funding from the Department of Health and Human Services, Office of the Assistant Secretary for Planning and Evaluation: `https://www.urban.org` (visited on 03/19/2021)

or particles. In the late 1950s Stan Ulam and John Von Neumann created the first cellular automaton for calculating the motions of liquids. The idea of this approach was to consider a liquid as a mass of single units and compute the behavior of each particle based on the status its neighbors. Cellular automata are often applied in studies of social interaction, for example Conway's Game of Life [Gar70] or Schelling's segregation model [Sch69] [Sch71].

While artificial intelligence (AI) was first focused on individual cognition, in the 1980s and later with the growth of the Internet, a rising interest in distributed AI, the study of the attributes of interacting artificial intelligence systems, emerged. One major technique of AI is machine learning [Mic83], enabling computer software to increase its capabilities by learning from past results. Learning models and evolutionary algorithms are useful for modeling populations which have the ability to adjust to new situations [GT05].

Simulation in the Social Sciences

This outline of the history of computer simulation shows that several methods of contemporary social simulation originate in natural sciences like physics. Although the research topics of social and natural sciences differ substentially, some approaches can be adapted. There are phenomena, which are unique to social sciences and the application of computer simulation to conduct scientific research in the field of human societies needs to be discussed. One important pattern of social simulation concerns the consequences of interacting agents. When the agent's behavior is described by simple rules, the combined behavior of the society of all agents can form highly complex patterns [GT05]. This phenomenon is defined by the term *emergence* and due to its importance, emergence and its implications for social simulation is discussed in greater detail in Chapter 2.2.6.

A central theme of social simulation is the study of nonlinear systems, which appear chaotic or unpredictable in contrast to simple linear systems. Conventional statistical methods for investigation of social systems are based on linear relationships between variables, where changes of the dependent variables are proportional to changes of the independent variables. This approach is restrictive and leads to limited models. The real world can only be considered as a nonlinear system. In economic systems, nonlinearity emerges from both human behavior and from the mutual influence of economic variables. Nonlinear system models cannot be solved analytically. Assuming a set of nonlinear equations, it is not possible to formulate the equation to be solved as a linear combination of the unknown variables. The effective method of investigating nonlinear behavior is to build a model of the system and simulate it. When simulating complex, nonlinear models, the explanation of a theory does not imply to make successful predictions. Complexity theory stated that even with a perfect understanding of the individual behavior, it is not valid to predict societal or group behavior [GT05].

Another characteristic of social simulation originates from biology and has been adapted in social science research because it describes the character of human interactions as reflexive. This characteristic is called autopoietic, or self-organizing. Autopoietic systems have to ability to reproduce and maintain themselves. An array of processes creates components that in turn recreate the processes that produced them. Autopoietic systems and human societies were theoretically analyzed by Maturana and Varela in *Autopoiesis and cognition: The realization of the living* [MV80], in which they define living systems as self-referencing, self-contained entities, instead of objects of observation and description [GT05].

Two trending topics that are increasingly recognized in social sciences are the theory of rationality and the addition of a spatial dimension. Rational choice theory was researched by Elster [Els86] and can be applied for simulation models using methods from the area of artificial intelligence. Often, the goal is to examine the consequences of rationality. One application of this theory is the modeling of markets where agents have limited, or local, information, similar to real world markets [GT05]. The first steps towards spatially-aware simulations were achieved by cellular automata, where agents interact on a grid of cells. More sophisticated simulations take the location of agents into account, which affects their decisions and behavior. To reproduce actual terrain, some simulation software tools feature graphical visualizations (see Chapter 2.2.8). This approach is used in the present thesis regarding the simulation of geospatial aspects of street crime.

2.2.2 Computer Simulation as Scientific Method

In the previous chapter, the real world phenomenon to be investigated by the social science researcher, has been introduced as the *target*. In social simulation, the target is not a static entity, but constantly changing and reacting to stimuli. The target features both state and behavior, so the model must be dynamic too. The model can be formulated and represented as a specification, for example mathematical expressions or a computer program. In order to draw conclusions from the specification, the model is analyzed how it behaves over time. There are two main approaches to derive future states of a model: an analytical method and simulation. The first approach means solving the problem statement by mathematics or logic. A frequent application of analytical methods are macroeconomic models, described by a set of equations. In this case, algebra is used to compute the results when certain variables change over time. But a numerical solution is not feasable, or possible, for all problems. As a result, the simulation approach is widely used for nonlinear models and complex specifications [GT05].

Simulation as a scientific method is derived from statistical and mathematical methods and follows very similar principles. In statistical modeling, social processes of the real world are observed and a model is formulated by abstraction of the target. This involves both the estimation of variables of the set of equations and the application of gathered data from the target to perform the calculations, which produce the predicted data. The results are analyzed in two steps. First, the predicted data is assessed whether the underlying model generates valid output, which corresponds to the observed data. Second, the parameters are evaluated regarding their magnitude. This approach also applies to simulation models with minor changes. Again, a model is built based on the observed target. But this model is formulated as a computer program instead of mathematic equations. The model is run on a computer and the output is recorded. The resulting data is compared to the target in order to verify that the model behaves similar enough to the real world processes [GT05].

While the parallels between statistical and simulation methods are evident, there exist major differences. Statistical models focus on investigating correlations between variables and in contrast simulation models focus on exploring systems with behavior. Hence, simulation models are able to not only reveal relationships, but also to discover the underlying patterns of these relationships. Both statistical and simulation approaches can be used for explanation and description. These two applications are not exclusive, but rather contribute to each other. A valid predictive model can also be used to gain a deeper understanding, while a valid explanatory model can function as basis for coarse or simple predictions [GT05].

The Stages of Simulation-Based Research

In order to make valid predictions or explanations of a phenomenon by a computer simulation that satisfies scientific standards, the research process must follow an established methodology. These important stages of simulation-based research are widely discussed in the available literature [Dor97] [Gil10] [OR10] [Ore+94] and refine the previously presented statistical approach. After a short introduction in the following paragraph, the methodology is discussed in greater detail.

The process starts with the formulation of one or more research questions, the goal of the investigation. A sample question could be: *Are criminals more susceptible to be victim of a crime than law-abiding citizens?* Then the target must be defined for the model, for example: *aspects of street crime in Maputo*. Next, observations of the target are required to develop a model. Continuing the example model, the gathered information consists of *geographic data* including the city street network and routine activity information defined as home, work and recreational places, *census data* as attributes of citizen agents like income and criminal propensity, and global variables like employment or crime rates. This information, for example obtained from a governmental agency, sets the initial conditions and the behavioral parameters for the simulation. Building a model also requires to formulate assumptions about the research goal. In the next step, the model is implemented as an executable computer program, which in most cases is developed using a simulation software package. After the implementation, the simulation itself is conducted by running the program and the output data is recorded. At this point, it is not ensured that the model is correctly programmed and behaving like the target. It is crucial for the scientific validity of the simulation to perform a thourough *verification*, *validation* and *sensitivy analysis*. Essentially, the *verification* step is necessary to ensure the correct implementation of the model. Like in general software engineering, this debugging step grows in difficulty with the complexity of the software. An aggravating factor is the use of pseudo random numbers in social simulations to mimic unknown variables and random events, so repeated runs are non-deterministic. The goal of the *validation* procedure is to guarantee that the behavior of the model corresponds to the behavior of the target. If crime patterns in a million city are being modelled, the simulation must reproduce to a determined degree the interactions of citizens as criminals, victims and law enforcement officers. Finally, the *sensitivy analysis* shows how the simulation model reacts to slight alterations of the parameters and initial values [GT05].

When designing a model as a simplification of the real world, the challenge is to decide which aspects of the target to include or to deliberately omit. A model that tends to be too simple means a greater conceptual leap concerning the conclusions learned from the model relative to the target. The more aspects are put into the model, the higher are the demands concerning the accuracy of the measured or assumed parameters. The goal is to develop a model that contains a minimum number of assumptions, but works for most situations depending on available data and research demands. In general, a more detailed model is beneficial for prediction purposes, while a simple model is better suited for explanation. A more complex model is not necessarily a better model. A high amount of data and high complexity render the process of verification and validation difficult to achieve and endanger the validity of the conclusions drawn from the simulation [GT05].

The next step after model design is its implementation. One way is to write a custom computer program that includes the complete simulation environment. Another method is to use an existing simulation toolkit to simplify the development of a simulation program. A simulation software toolkit already covers frequent use cases like data input and output, visualization, multi-agent support, a user interface to start and stop the simulation and more. Naturally, the usage of a standard simulation software entails limits concerning the development and the available styles of the simulation. In the case of a custom development without the use of a simulation software, the following considerations regarding the programming language are relevant. Simulation development is exploratory and cyclic, so the language should support agile development cycles for coding, debugging and refactoring. Other beneficial factors are debugging capabilities and library support. Simulations generate lots of data, depending on number of recorded agents, attributes, duration of the simulated period of time. For example, a simulation models 1000 agents and their interactions during one year. The output is a record of 20 agent attributes on every simulation step (tick), where one simulation tick corresponds to one minute in the target, so one year amounts to 525 600 ticks. The data is assumed to be encoded as 32-bit Integer. The calculation results approximately 42 048 Gigabytes of data. Another crucial factor is the execution speed of the simulation. A sensitivity analysis requires repeated simulation runs, so the simulation program needs to run efficiently, or else it is not possible to successfully complete the research regarding time resources and scientific quality. In practice, various programming languages and environments are suitable for simulations depending on requirements, but interpreted languages like Python, R, MATLAB, and compiled languages such as Java, C, C++, Smalltalk, Prolog and Lisp are reported to be commonly used [GT05].

In the presented methodology, the verification step is to verify that the computer program is working as intended. In an agile development environment, verification is a cyclic part of simulation development and supports incremental refinement and rapid prototyping. The complexity of the program and the application of random number generators are relevant factors for the verification process. The use of random numbers causes different output data on every simulation run, so only the distribution of the results can be anticipated. Standardized test cases that cover certain predictable scenarios are a well-suited tool for verification. The test cases can be executed after major code changes to avoid new errors [GT05]. Simulation toolkits support program verification by for example providing repeatable random number generation, prepared test case methods and built-in version control.

While the previously discussed verification measures are concerned with software quality, the validation aims to ensure that the simulation is a well-suited model of the target. The model is expected to reproduce the behavior of the target according to the research questions. The validity of the model is asserted by comparing simulation results with observed data from the target. Since model and target both at least partly underly stochastic processes, a precise correspondence cannot be expected. Whether simulation results that deviate from the collected target data indicate a non-viable model, is depending on the defined statistical distribution of the results. Another relevant aspect is the path-dependency of many simulations. Path dependency means that „current and future states, actions, or decisions depend on the path of previous states, actions, or decisions“ [Po06]. Accordingly, simulation results are the consequence of certain input conditions and subsequent stochastic events during the simulation. When evaluating the model, it is possible that the model is valid, but the observed data about the target is flawed or itself a product of estimates and assumptions. One specific difficulty concerns highly abstract models. In this situation it is hard to correspond output data from the model to specific data from the target [GT05].

The next step after validating the model is the sensitivity analysis. This procedure aims to investigate how certain variables influence the model results. It is a common practice for discovering the most important parameters for system behavior on which the simulation effort should focus on. The sensitivity analysis comprises of varying model parameters systematically and observing how the simulation results change. One parameter at a time is varied, while the values of all other parameters remain constant. That way, the researcher is able to account for uncertainty regarding chosen input parameters, or gain an improved insight on how the variable affects the system behavior. This knowledge, for example identifying best- or worst-case scenarios, is critical to modeling [NM07]. Another application of sensitivity analysis is to certify the robustness of the model. If the behavior is too sensitive to small changes in the value of the variable, the correctness of these values should be re-evaluated [GT05].

The publication of the simulation results marks the final step in simulation research. The documentation of the research should be detailed enough to allow the audience to understand and reproduce the simulation. The length of a typical journal article is often not sufficient to describe and document the simulation in every detail. One common solution to this dilemma is to publish the source code and supplemental data such as input data, result data, examples, UML[17] diagrams or unit tests on the internet for download [GT05].

Simulation versus Experiment

Simulation as scientific method is similar to an experimental methodology. It can be executed and observed repeatedly, while the parameters are varied to explore the behavior of the model. The difference is that experiments involve the actual real world object, while simulations analyze the model as an abstraction of the real world. This view centers around the idea that experiments primarily give insights to the real-world phenomenon and only secondarily allow deductions to the behavior of related systems. For example in biology, studies on tulips teach us about tulips in the first place and only secondarily allow inferences about plant genetics in general. This approach differs from computer simulations, which are not primarily supposed to center around the behavior of computers, but rather provide findings about the simulated model. Another view emphasizes the similarities. The targets of experiments also act as surrogates of the explored phenomenon and most experiments require comparable modeling techniques as simulations. Parker [Par08] showed in his research of the epistemology of computer simulation that there are relevant similarities between experiments and computer simulations. Given enough knowledge, a simulation can produce a greater insight into a system than an experiment [Win19].

2.2.3 Types of Computer Simulations

Generally, two types of computer simulations are distinguished: equation-based simulations and individual-based simulations. Both types support the most common applications of computer simulations as introduced in Chapter 2.2.1 [Win19]. The most relevant equation-based model is *system dynamics*. *Microanalytical simulation*, *cellular automata*, *agent-based models* and *evolutionary models* are representatives of individual-based simulations. This chapter provides a concise overview of the aforementioned simulation techniques. Agent-based simulations, being the main focus of the presented research, are discussed in greater detail in the Chapters 2.2.4 and 2.2.5.

Equation-based simulations are typically applied in nature sciences, where a theory can be described by a mathematical model based on differential equations. This type of simulation is either particle-based or field-based. The former contains n discrete entities and a set of differential equations describing their behavior. For example a simulation of the sun system, where the gravitational effects of the planets and the sun are investigated. Field-based means that the set of equations is controlling the behavior of a continuous field over time. An example of this type of simulation is a fluid model, such as a meteorological system. *Individual-based* simulations are primarily used in social sciences, with other applications in disciplines like behavioral sciences, artificial intelligence, epidemiology, or

[17] Unified Modeling Language (UML) is a modeling standard in software engineering, maintained by the Object Management Group (OMG). Online at `http://www.uml.org/` (visited on 03/19/2021)

ecology. These disciplines have in common that the interactions of many individuals in a network is a typical research focus. Compared to particle-based simulation, individual-based simulation also investigates the behavior of n entities. The major difference is that individual-based simulations do not contain global differential equations that control the actions of the individuals, but a set of rules at the individual level, which determines their behavior [PSR98] [Win19].

System Dynamics

System dynamics [Har91] [Ste00] historically evolved from equation-based simulations, where the target system is described by difference or differential equations. This system of equations enables the computation of future states of the target from the current state. The target is simulated on the macro level, described by a set of attributes, which determine the state of the whole system and its transitions. The basic difference equation can be written as

$$x_{t+1} = f(x_t; \vartheta)$$

where x_{t+1} is the system's state at time $t + 1$, which is derived from the initial state at time t and the parameter ϑ, a vector of n elements. f is a continuous function. In comparison, a differential equation has the form

$$\dot{x}(t) = \frac{dx}{dt} = g(x(t); \vartheta)$$

where $\dot{x}(t)$ is the first-order derivation by time dt. The transition depends on the state $x(t)$ and the vector ϑ. Difference and differential equations are closely related, though the solutions differ. Only the simplest difference and differential equations can be solved explicitly. The procedure to find numerical solutions makes use of the similarities of both types and a fixed time scale Δ_τ. This is the main principle of system dynamics. Compared to a system of differential equations, system dynamics use discrete time as approximation of continuous time and they are not restricted to continuous functions [GT05].

System dynamic models are widely used in engineering, economics and also social sciences. They can be interpreted as causal loop diagrams with stocks, flows, feedback loops, time delays and table functions, but allow researchers to study the behavior of the target. DYNAMO, developed in the 1950s at M.I.T. Computation Center to simulate Jay Wright Forrester's World Model, was the first language designed for building system dynamics models [For71]. Since then, other products for simulating system dynamics have been developed, for example STELLA[18], PowerSim[19] or VenSim[20] [GT05].

[18] https://www.iseesystems.com (visited on 03/19/2021)
[19] http://www.powersim.no (visited on 03/19/2021)
[20] http://www.vensim.com (visited on 03/19/2021)

Microanalytical Simulations

While system dynamics only describe its targets as indivisible totals, microanalytical simulation models specifically are concerned about the individual level. This is a consequence of the fact that targets in social sciences generally consist of individual entities, such as persons, households, subpopulations, or companies. As a result, a different modeling approach that incorporates at least an individual and an aggregate level, has been developed: the microsimulation method.

One main application of this simulation method is to predict individual and group behavior caused by certain policies at the aggregate level. Another relevant application concerns the studies of demography. When using a system dynamics model in order to investigate the changing age structure of a population, only the macro level is simulated. This approach would work for basic research of the age structure. But when the research is interested in more details, for example fertility rates for certain education levels, the system dynamics model would grow too complex. These types of research problems are better suited for microsimulations, which model individual entities with a set of attributes and a set of transition probabilities. As a result, this approach is a stochastic model, in contrast to the deterministic system dynamics approach [GT05].

One drawback of the microsimulation method is the required level of detail concerning the data of the initial state at the individual level. They are very demanding towards data collection, data processing and data storage capacity. The computing time c often is proportional to the number of simulated individuals x to the square ($c \in \mathcal{O}(x^2)$), while the storage usage S is a linear function ($s \in \mathcal{O}(x)$). Historically, these high demands restricted microsimulations to large and expensive mainframe computers, operated by specialists, during the 1970s and 1980s. There were no widespread microanalytical simulation packages available, so models were developed separately in general-purpose languages like FORTRAN or PL/1. Today, there is a diverse landscape of microsimulation packages for many appliances [GT05].

The available literature describes different methodologies for microanalytical simulations. Depending on how the population data is changed during the simulation, microsimulations can be static or dynamic. *Static* microsimulations change their population data by adding updates from external sources. The population is aged by modifying weights of its attributes, producing a new file for the future year. This type of simulation is used for short-term predictions of effects of a certain policy, like the impact of tax rate changes. It can support decision-making by testing hypotheses about the consequences of policy changes based on current and known characteristics of the population. For example, if the tax on good X was substantially increased, people might buy the substitute good Y instead. The limitation of the model is that impacts of policy changes cannot be evaluated based on forecasts of future economic and demographic characteristics of the population. *Dynamic* microsimulations overcome this limitation by computing a new state of each entity individually on every time step. This way, individual attributes and activities, or life processes such as aging, employment, movement, acquisitions, birth and death are specifically modelled. This approach enables dynamic microsimulation to study long-term effects of demographic change [GT05]. A famous microsimulation model for investigating governmental tax, transfer, and health programs in the United States is TRIM3 [Urb12], developed and maintained at the Urban Institute[21]. Other representatives for economic and financial decision-making are Euromod[22] and Pensim2[23]. In the area of traffic simulation research there exists a broad variety of software packages, for example TransModeler[24], PTV Vissim[25] and TSIS-CORSIM[26].

Cellular Automata

Cellular automata (CA) are built of a large grid of cells like a chess board. Each cell is assigned one state and a set of rules define when changes between states occur. Typically, the underlying rules depend on the states of the cell's neighbours. Simulations with complex behavior can be generated by very simple rules. The main research focus lies in modeling social phenomena with emerging characteristics from interaction at the individual level. Classic applications are the modeling of the dissemination of information, the formation of cliques, social segregation and life-like cellular automata. A cellular automaton exhibits five special features, differentiating this method from other simulation types. First, the CA consists of a grid of cells. Typically, the grid is rectangular with N rows and M columns, but one-dimensional and three-dimensional environments are also possible. Each cell represents an entity at the individual level, for example humans, animals or households. The second feature defines the state of the cells. Each cell must be in one of several states, which represent certain characteristics of the observed individuals. The time component is the third attribute of cellular automata. Time is simulated by advancing the model in steps. At every time step, the state of all cells is evaluated. Fourth, at each time step the state of a cell is determined by its rules. The new state at time $t+1$ is computed from the previous state at time t as a function of the states of the cell's direct neighbors. All cells share the same rules. The last characteristic concerns the cell's neighborhood. Cellular automata simulate local interactions like the spread of information or diseases [GT05]. A very basic, but well-known example of cellular automata is John Horton Conway's Game of Life [Gar70].

In Conway's Game of Life, neighborhood is defined as the eight cells that surround one cell, which is also known as Moore neighborhood. The cellular automata consists of a grid of cells with two states: alive or dead. A cell survives, if there are exactly two or

[21] Urban Institute, with primary funding from the Department of Health and Human Services, Office of the Assistant Secretary for Planning and Evaluation: `https://www.urban.org` (visited on 03/19/2021)

[22] Euromod is the tax-benefit microsimulation model for the European Union: `https://mctrans.ce.ufl.edu/featured/tsis` (visited on 03/19/2021)

[23] Pensim2 is a dynamic microsimulation model for predicting future pensioner incomes between 2006 and 2050: `http://www.pensionspolicyinstitute.org.uk` (visited on 03/19/2021)

[24] `https://www.caliper.com` (visited on 03/19/2021)

[25] `http://vision-traffic.ptvgroup.com/en-uk/products/ptv-vissim` (visited on 03/19/2021)

[26] `https://mctrans.ce.ufl.edu/featured/tsis` (visited on 03/19/2021)

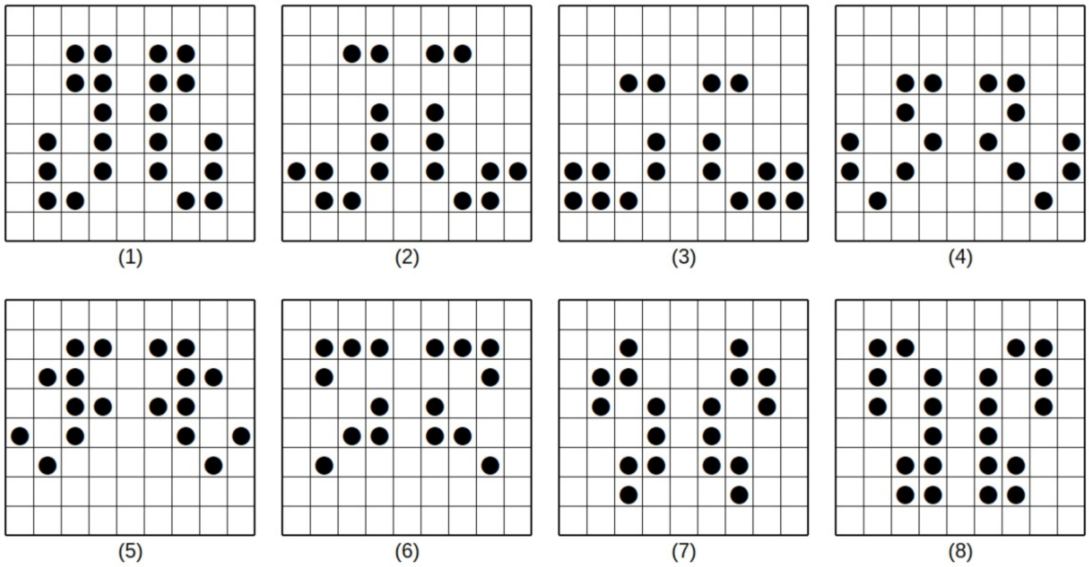


Figure 2.13: The evolution of a Game of Life pattern over the course of eight time steps.

three living cells in its Moore neighborhood. If there are more than three living cells in its neighborhood, the cell dies of overcrowding, because too many individuals share limited resources. Similarly, if there are zero or only one living cells in its neighborhood, the cell dies of loneliness. This behavior describes the first rule of the Game of Life. The second rule states that a dead cell bursts into life if it has exactly three living neighbors. Those two trivial rules generate a multitude of changing patterns when the simulation is run. Figure 2.13 shows the evolution of a pattern over eight time steps. The eighth pattern is an inverted version of the first and the fifteenth would be the same as the first, because the pattern repeats every 14 steps. There are many other patterns, some cyclic, some reproducing, or with other interesting features [Gar70] [GT05].

Many other cellular automata models have been developed. They differ in the rules to change the cell's state and in the definition of neighborhood. In addition to the previously defined Moore neighborhood, another relevant type is the von Neumann neighborhood. It is defined as the four adjacent cells to the north, south, east and west. The parity model is one application of the von Neumann neighborhood. This model features only one rule. The cell's state is evaluated either alive or dead depending on whether the count of living cells in its von Neumann neighborhood is odd or even. The gossip model simulates the spread of knowledge from a single originator to his neighbors, who in turn pass the information to their own neighbors. Each cell is an individual with two states, knowing the gossip and no knowledge. A cell can only change its state from not knowing to knowing when at least one of its four von Neumann neighbors know the information. A cell can never forget the information, so it is not possible to change the state back. While the previously discussed models Game of Life and parity model are deterministic,

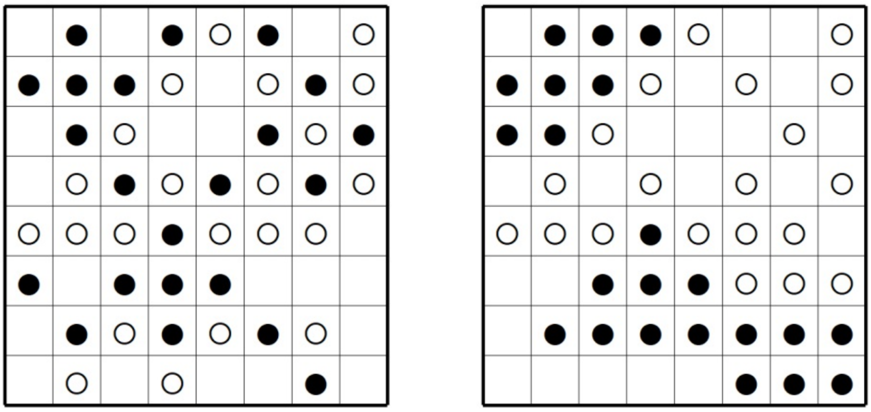


Figure 2.14: Left figure shows the initial condition of Schelling's segregation experiments; right figure shows a stable segregated pattern obtained in several iterations [Sch71].

the gossip model is stochastic. A random number generator decides whether a cell will pass on the gossip to its neighbor. The model shows that even a very small probability does not prevent the transmission, but only leads to a slower diffusion. So far, all models featured fixed individuals. One important advancement is the migration model, which enables actors to move around the grid [GT05].

Thomas Schelling published two widely recognized papers in 1969 [Sch69] and 1971 [Sch71] describing a segregation model of households in the United States of America. This migration model followed the idea that people have a limited tolerance regarding the presence of other ethnic groups in the neighborhood. They are content as long as a certain amount of neighbors is of their own group, but move to another neighborhood otherwise. For example, a tolerance threshold of 20 % means that people are content and do not move as long as at least one out of five neighbors is of the the same ethnic group. Schelling showed that even tolerance thresholds lower than 50 % lead to ethnic segregation. His model consists of a grid of cells with three states: white actor, black actor, or empty. At every time step, all occupied cells are evaluated. The inhabitant is analyzed whether it is content with its neighbors. If this is the case, the next occupied cell is selected. If not, the actor moves to a neighboring cell that is both free and features a neighborhood that the actor would be content living in. The simulation terminates when all actors are content. Figure 2.14 shows the initial condition of Schelling's segregation model with black and white actors distributed randomly on the left hand side. The right hand side shows a stable segregated pattern obtained in several iterations, where all actors are content [GT05].

Learning Models

The presented simulation methods in this chapter have in common that the models remain unchanged during the simulation. This characteristic is different for learning and evolutionary models, which have the ability to adapt the underlying model or its parameters in response to the environment. Two relevant approaches of learning models for simulation of social phenomena are artificial neural networks (ANN) and evolutionary algorithms. Both techniques are inspired by biology.

The human brain consists of billions of neurons, connected by synapses in order to form a communication network. Neurons communicate with each other by means of electrochemical signal pulses. If a neuron receives a excitatory input from another neuron, it sends a signal to certain other connected neurons. A learning process occurs when two neurons send at the same time and therefore strengthens the connection between them. Artificial neural networks aim to imitate this behavior on a simplified scale. They consist of a network of nodes, or artificial neurons. An ANN is composed of three or more layers. The input layer receives input from the environment, the hidden layer transforms input signals into output signals and the output layer forwards the response of the network. Each connection, corresponding to synapses in the brain, can transmit a signal from one node to another. The artificial neuron receives the signal, processes the signal and fires a reply to other nodes. In ANN, a number is transmitted and the signal output is computed by a function depending on the set of input numbers. The edges between network nodes typically have weights. They are constantly adjusted during the simulation and represent the strength of the connection [GT05].

Evolutionary models apply the process of evolution by natural selection to computer simulation. They are based on genetic algorithms and aim to find optimal solutions to complex problems. One example is the traveling salesman problem (TSP) [Lar+99] of the class of NP-complete problems. In the case of the TSP, an analytical solution is not feasable due to factorial computation time $\mathcal{O}(x!)$, so approximation algorithms have been developed. Evolutionary algorithms are also applied to model changes within societies and in game theory [Axe97]. The genetic algorithm model aims to optimize a specific fitness measure by evolving a population over the course of several generations. This fitness measure of every individual of the population must be chosen carefully. Possible fitness measures of individuals are happiness, accumulated wealth, or other individual characteristics. The function to calculate the fitness measure should be efficient, because it is the most computationally intensive part of the simulation. In the first step of the algorithm, N individuals with random genes are generated. Then, the fitness measure is calculated for each individual. Those individuals with the highest measure are selected for mating. Pairs of individuals from the mating pool combine their genes to produce a new generation. A small fraction of the genes is mutated in this process. Next, the old population is deleted and the fitness measure is calculated for each individual of the new generation. The algorithm terminates when the fitness measure of the population is no longer improving. This algorithm is an optimization method, because those individuals with a higher fitness measure are selected for reproduction [GT05].

2.2.4 Foundations of Agent-based Modeling and Simulation

Several types of computer simulations have been briefly discussed in the previous chapter. Agent-based modeling and simulation (ABMS) takes a different approach from other techniques. Together with microanalytical simulations, agent-based simulations are concerned about the individual level. But where microsimulations are restricted in terms of agent behavior and agent communication, ABMS further developed this concept of individual and aggregate levels. Microsimulations are data-driven and often start from empirical or census data, while agent-based simulations tend to be concept-driven. One main limiting aspect of microsimulations is that agents at the individual level are not able to react to their effects at the macro level, which is one core principle of agent-based simulation. Basically, agent-based modeling centers around virtual agents with a set of instructions, which enable those agents to interact. As in microsimulations, these agents can be human beings, animals, wares, or any other discrete object. From the behavior of the agents, patterns emerge that can be quantified and explored in order to find reasons for the specific phenomena. These patterns on the macro-level result from the individual decisions of the agents at the micro-level [Joh13].

Agent-based modeling and simulation is a method to examine the effects of the behaviors of individuals at the system-level. The behavior of individuals is specified by certain rules and it is embedded by an environment in which the agents act. By application of these rules within the system context, the simulation generates agent behavior as system outcomes. Both *deterministic models*, where actions at the micro-level determine the repeatable results at higher levels, and *holistic models*, where stochastic functions create nondeterministic results, are supported. In contrast to microsimulations, the individual and the aggregate levels constantly influence one another in feedback loops. Within their set of rules, agents are able to make decisions based on adaptation and variability. In order to make such decisions, agents are restricted to a limited amount of time to consider their actions and process only finite information. This concept is called *bounded rationality* [Sim97]. Information available to agents is not only finite, but also local in a sense that each agent resides in a limited neighborhood from which the agent can acquire information from a limited number of sources [Joh13].

Another key concept of agents in ABMS, differentiating this method from other simulation approaches, is their ability to adapt or learn. The concept of learning from past actions may range from simple adjustments of single parameters to developing entirely new characteristics and strategies. Adaptation as a special form of behavior similarly requires rules to occur. Agent-based models without the feature of adaption are described as *proto-agent* models. The system in which agents act does not make decisions. The environment is not a static system, but may be modified over time. It can provide an information database for historic agent actions, which potentially affects future agent behavior. Agent behavior can be further characterized by the chronology of their actions, and by their tendency of reactivity. The first concept describes agent actions taking place *synchronously*, where each agent acts at discrete time steps, or *asynchronously*, where arent actions are triggered by a clock. The second concept characterizes agent behavior as either goal-oriented or reactive. For example, goal-oriented behavior occurs when agents perform specific actions until certain conditions are met. Reactive actions are triggered by the environment or by actions of other agents [Joh13].

The History of ABMS

The roots of agent-based modeling and simulation can be traced back to cellular automata (see Chapter 2.2.3), created by Stan Ulam and John Von Neumann in the late 1950s for calculating the motions of liquids. Schelling's segregation model [Sch69] [Sch71] is regarded as another foundation for ABMS [Joh13]. The theoretical background for ABMS was developed by the research of complex adaptive systems (CAS)[MW02]. This scientific area is engaged in the study of systems with the capability to adjust to a non-static environment [NM07]. In the 1980s Robert Axelrod investigated interaction strategies in game theory such as the prisoner's dilemma [AH81]. Axelrod tested several strategies using ABMS and discovered that more altruistic strategies performed better than greedy strategies.

In the 1990s Joshua M. Epstein and Robert Axtell presented a general-purpose agent-based model in their book „Growing Artificial Societies“ [EA96]. This model called *Sugarscape* is based on a two-dimensional cell grid, agents, and the rules that control the behavior of the agents [GT05]. Several open-source and commercial simulation software packages implemented the Sugarscape model, for example MASON [BCRL07], Wolfram Mathematica [Chr11] and NetLogo[27].

Since the year 2000 many agent-based models have been developed for social sciences. One notable model was built by Ron Sun, exploring the „intersection between individual cognitive modeling and modeling of multi-agent interaction“ [Sun06]. In the area of ABMS for criminology relevant research was conducted by Elizabeth Groff [GM08], Joshua M. Epstein [Eps02] and Nick Malleson [MHS10]. Macal and North provide an overview on historic social simulation models in their work *Managing Business Complexity* [NM07].

Applications

Agent-based modeling and simulation is commonly applied to investigate systems consisting of interacting individuals. Applications that match this definition exist in various areas. A selection of examples is presented in this chapter. Consumer markets are explored to research strategic solutions for enterprises. The agents are consumers and companies, which interact on markets and generate complex market effects. In the research of industrial supply chains, synergies are determined within systems of autonomous units. In the health-care sector, the agents are patients and medical personnel [NM07]. In biology, ecosystems are analyzed towards the contribution of individual members to the overall status of the system, for example the MANTA model [DCL95]. ABMS plays

[27] NetLogo Models Library documentation online at `http://ccl.northwestern.edu/netlogo/models/` (visited on 03/19/2021)

an important role in the research of social sciences. For example the EOS (Evolution of Organized Society) model [Dor+18] investigated the complexity of early human societies during the Upper Paleolithic period. Other applications in social sciences include aspects of human societies such as segregation, crime, employment and other phenomena. In this thesis, ABMS is applied in the context of social sciences to study the spatial effects of the routine activity theory by Cohen and Felson [CF79] on crime rates. In general, Joshua M. Epstein regards agent-based simulation as a powerful instrument in the study of „spatially distributed systems of heterogeneous autonomous actors with bounded information and computing capacity“ [Eps06]. Robert Axtell identified three roles for agent-based simulations in social sciences: the ABMS version of an equivalent solvable numerical model, an approach for investigating partially unsolvable equation-based models, and as a substitute for analysis for formally undecidable or intractable models [Axt00].

The main feature of agent-based modeling and simulation is the agent perspective, differentiating it from other simulation types. ABMS neither starts from large statistic or census data, nor does it start at the aggregate level. It begins at the individual level, where decisions are made, and builds top-level patterns from there. The essence of agent-based modeling is to describe the behavior of single members of a group, because it is not possible to define the behavior of the whole system in advance. The computer runs the model many times, computing agent interactions and thus generating emergent patterns. Agent-based models can reveal the evolution of a system over time, which is often hard to forecast from knowledge of the characteristics of the individuals alone. It can be applied to gain an understanding of aggregate-level behavior that is the result of complex interactions of agents. The iterated application of individual interactions sometimes creates unpredictable patterns on both the individual and the system-level, establishing a link between the micro- and the macro-levels of the system [NM07].

Adding a Space Dimension to ABMS

One advantage of agent-based modeling and simulation is the support for extending the model by a space dimension. The spatial relationships in ABMS can be classified into four different aspects of agent bahavior. Agent movement through the environment is one of them. The individual relocates within the system by an available type of movement in a time step. The second aspect concerns the decisions that agents make in relation to space. For example, agent behavior can be affected by its own location, the location of other agents that the agent interacts with, or by its environment. The third aspect describes the individual's ability of changing or re-arranging the environment. This leads to the fourth facet that states that the individual's decision-making is subject to change with the changing environment. In order to simulate agent bahavior based on spatial relationships, spatial analysis is required. These computations, including the measurement of distances, clipping or overlays, are usually done by spatial modeling software such as presented in Chapter 2.1.2 [Joh13].

Randomness in ABMS

In agent-based simulations, random numbers are introduced to model external processes, or exogenous factors that are not part of the model. In this approach, a stochastic function provides values for uncertainty, or for parameters that are impossible to measure. An example for randomness in ABMS is the modeling of a customer queue in a restaurant model. The phenomenon of uncertainty describes the absence of information, like the point in time of the event when the next customer decides to queue up [Bun09]. Randomization of parameters can be applied to extend agent rules for the simulation of human-like behavior, for example emotions or for subjective decision-making. There are agent-based models where the results are dependent on the exact order of the agents' actions. In order to avoid these effects, the sequence is randomized. In these cases of randomness contributing to the model, the model is simulated repeatedly to generate results in various conditions. The simulation outcomes are treated as distributions, or alternatively as means with confidence intervals. Once a random element has been introduced to the model, the same statistical methods as for experimental research apply, where quantitative changes are evaluated by regression and qualitative changes are evaluated by analysis of variance [GT05].

2.2.5 Agents in the Context of Social Simulation

In the previous chapter, the foundations of agent-based modeling and simulation have been elaborated. The agent-based model has been defined to consist of three parts: the agents, the set of rules they act upon, and the environment in which the agents exist. This chapter focuses on agents and discusses their properties, behaviors and architectures. Agents are decision-making entities within complex adaptive systems, controlling their behavior based on their perceived environment [GT05]. All individual entities that make choices qualify as agent. The behavioral rules enable agents to receive information, process that information and undertake action accordingly [NM07] [SM06].

Properties of Agents

In order to discuss the characteristics of agents in the context of social simulation, the idea of *agency* should be introduced. In social sciences, agency represents the ability of individuals to act and decide independently. For agents as computer programs, this concept applies in a restricted scope. The computer agent neither has intentionality, nor some degree of free will, but it is programmed to simulate simplified properties of human behavior. Five general properties are referred to computer agents. From these general properties, a set of specific properties and attributes are derived in the following section. Agents exhibit *autonomy*, which enables them to act without being controlled by an external force. Agents also possess a *social ability*, allowing agents to interact with other agents. In ABMS, this interaction occurs by application of a computer language. The third general property is *reactivity*. Agents have the ability to identify external stimuli from other agents or from the environment and act accordingly. The correspondent attribute to reactivity is *proactivity*. This property enables agents to act on their own initiative towards a defined goal [GT05]. Agents are also adaptive, which gives them the ability to learn and change their behaviors according to their past experiences [NM07].

Agents have a form of knowledge; they act based on information about their environment. This information is incomplete and sometimes even incorrect. Based on their knowledge, agents may have the ability to derive further information. Another property concerns interaction between individuals. Agents may recognize relationships of other agents and build a social model based on this information. Similarly, agents may build other views of their environment based on their existing knowledge and perception of the world. The concept of knowledge representation and reasoning is a field of artificial intelligence research and a key factor in agent-based modeling. One approach is the application of predicate logic as deductive system. Another approach to gain knowledge from stored information uses semantic networks. This is a knowledge base consisting of concepts and semantic relations forming a directed or undirected graph [GT05].

A broad array of attributes is modeled in ABMS. Common attributes of agents representing human beings in social simulations are age, sex, employment status, income, various preferences, or the current location. Simple attributes, like age or name, can be stored in one-dimensional variables. Other attributes, like preferences, are multifactorial and require more complex, nested definitions. A relevant property of attributes in agent-based modeling and simulation is that they are often subject to change over time based on the agent's past behavior [NM07].

Agent Behavior

Agents feature various behavioral patterns, foremost the ability to adapt, and decision routines to select the course of action. Other common behaviors involve social interaction and sensoric functions to perceive the environment. Decision routines are higher-order rules that enable adaptation by permitting the modification of base-level rules. Agent behavior is modeled in three steps. Initially, agents analyze their state and execute their decision-making algorithm. After the calculation, the resulting action is executed. In the last step, agents examine the consequences of their action and modify their rule set based on the outcomes [NM07].

Agent behavior has also been characterized as autonomous and proactive. Combining these properties, agents have the ability to independently act towards a specific goal. A classic example of such a goal is survival. This broadly defined goal consists of several secondary goals, such as satisfying nutrition, resting regularly and avoiding danger. One challenge for the agent is to prioritize various goals, which may be of different relevance and may even conflict each other. For example, the search for nutrition consumes energy and puts the agent in danger, but without enough nutrition, the agent would starve eventually. One solution to this problem is the implementation of a decision planner module as integral part of the agent. The task of a decision planner is to determine the behavior that maximizes the degree of satisfaction of the agent's goals. This behavior ranges from simple *IF state X THEN act Y* conditions, to complex planning like temporarily moving away from one goal in order to reach another, more important one [GT05].

Social behavior has been introduced as interaction between agents. This interaction is realized by exchanging messages that contain information. Agent interaction must not necessarily be intentional, it may be a side effect of an action. While people use a natural language for communication with other people, agents usually exchange a kind of computer language, since modeling human language is a considerable challenge and out of the scope for most ABMS. Another concept attempts to model human emotions, like anger, happiness, or affection. There is an unresolved debate whether agent emotions are simply attributes or emergent entities of agent behavior. Emotions and the achievement of goals are directly related. One point of view regards emotions as an indicator of the status of the agent's decision planning. For example, a suboptimal emotional status effects a strategic change in decision-making in order to improve its emotional status. Another school of thought considers emotions as being only of secondary consequence to agent behavior. Emotions are seen as indicator of the agent's status within the system, but they are not conceded to play a causal role in the decision-making process [GT05].

Agent Architecture

The symbolic paradigm [NS76] is a traditional approach to artificial intelligence and a basic method to constructing agents with cognitive behavior. According to this paradigm, an individual is capable of intelligent behavior by manipulation of symbols. Symbolic AI uses production rules, which link symbols that are human-readable expressions, in *If-Else* conditions. There are three significant problems concerning the symbolic paradigm approach. The first addresses the phenomenon that slight variations of symbols often cause errors in symbol processing and reasoning. Second, there is a high complexity inherent to algorithms that compute decision planning tasks. Finally, it is challenging to represent common-sense knowledge. A range of techniques have been developed to address these difficulties. The two most relevant for agent-based modeling and simulation are production systems and object orientation [GT05].

As already presented, an agent-based model consists of agents, rules and an environment. A simple prototype of a rule system is a production system. It is composed of rules, a storage and a rule interpreter. A single rule consists of a condition clause and an action clause. The condition defines when the rule is fulfilled. The action determines the behavior in that case. The storage, or working memory, persists facts such as attributes and states of the agent and knowledge about other agents and about the environment. The rule interpreter evaluates each rule and determines whether the conditions are fulfilled. When the condition clause of a rule is met, the rule interpreter initiates the corresponding action defined in the action clause. One benefit of the presented rule system is that it is not necessary to define an arbitrary order, in which the rules are executed. This would be the case with common imperative programming languages or flow charts. The selection of rules and the time when rules fire are dependent on the stored facts in the working memory. One task of the modeler is to specify the course of action when more than one rule is fulfilled and ready to fire. This can be achieved by the introduction of a conflict management algorithm, especially in the case of conflicting general and specific rules [GT05].

The application of an object-oriented programming paradigm is the predominant style of developing agents. Various types of agents can be addressed by the concept of inheritance and individual agents are created by instantiation of the defined classes. The attributes encapsulated within a class object correspond to the agent's states, working memory and defined rules. The methods of a class perform as rule interpreter and the agent's behavioral actions. The advantage of the object-oriented approach is that all agents instantiated from a class share the same properties, actions and rules, while the individual values stored in attributes may differ [GT05].

The third component of an agent-based model is the environment. It provides a spatial context for the simulated agents. The characteristic of the environment, where agents are located, depends on the scope of the simulation. The most common environment is a grid, as already presented in cellular automata, where the position of an agent is defined by a tuple of (X, Y) coordinates. If elevation is relevant for the simulation, a Z coordinate is added. Another form of spatial representation is a network consisting of edges and nodes. Common spatial use-cases in ABMS are for example that agents are able to move through the environment, or that agents interact with each other based on their current location by a means of communication. In order to perceive their local environment, agents need sensors that provide them with information for their working memory. The concept of agent interaction exhibits a concurrency problem. In an ideal model, all agents act in parallel. Generally, it is not possible to simulate this exact behavior given a limited number of processing units, or threads of execution. As a result, agent actions are computed sequentially, and simultaneous behavior must be simulated. There are seveal techniques for choosing the sequence of the agents to act, for example a naive round robin approach, or a random function. The order in which agents operate may have a significant effect on the outcomes of the simulation. One basic strategy is the implementation of a message buffer. In the first step, all agents send messages sequentially and they are stored within the environment. In the second step, all messages are retrieved and delivered to the agents [GT05].

Agent Complexity

Agency has been introduced as a number of characteristics that describe agents, for example the ability to act autonomously, the ability to learn and adapt, or the persuit of goals. Not all models include all of the presented defining characteristics, but still qualify as agent-based models. If agents miss one or more features of agency, the agents are considered prototype agents, or proto-agents. This is often the case when a model is developed incrementally and more characteristics are implemented at a later development cycle. It may be intentional to omit a subset of these characteristics, because the agent-based model does not require them. The simulation is then considered a proto-agent model. Proto-agents generally consist of basic decision rules, which respond to the environment, or interact with related agents. In this concept, reactive rules correspond one external trigger to one specific response. Proto-agents fire one specific rule during each decision-making process. Complex agents in contrast are not restricted by this behavior. The rules of proto-agents are very simple, which do not show adaptive behavior per se. Adaption can be implemented in proto-agents by configuration of the agents. The interaction pattern of the agents are modified in reaction to alterations of the environment. This kind of adaptation of proto-agents occurs at the system level, not at the individual level like the adaptation process of complex agents [NM07].

Proto-agents consisting of simple reactive rules have several advantages, like centering on core mechanisms, less development effort, reduced verification and validation complexity, and fast user adaption. The focus on core mechanisms allows users to concentrate on the essential interactions and behaviors of the model. Complex patterns sometimes conceal the substantial rules that generate system results. Generally, the application of fewer rules improves the study of the contributions and consequences of each rule to the whole system. A useful agent-based model demands a sophisticated balance between abstraction, simplification and level of detail. It is considered best-practice to initially implement a simple model with few and basic rules and afterwards following an iterative development cycle to further refine the model [NM07].

Another advantage of keeping agent rules simple is the reduced development effort. Basic models can be implemented in less time and in a more economic development process. Basic models offer a solid groundwork for later extensions and help avoiding errors in early development phases. The verification procedure is necessary to ensure the correct implementation of the model. The goal of the validation method is to guarantee that the behavior of the model corresponds to the behavior of the target. The concept of verification and validation are described in more detail in chapter 2.2.7. Like in general software engineering, this debugging step grows in difficulty with the complexity of the software. This important step of quality assurance requires less effort for models with simple rules than for models with complex rules. The reasons are that proto-agents with simple rules lead to smaller ranges of system results, are easier to analyze and result in more compact program code than sophisticated models. One disadvantage of simple rules regarding verification and validation is that, under certain conditions, it can be more difficult to correspond such simple models to real-world phenomena due to the high degree of abstraction [NM07].

The final advantage of models consisting of simple rules is that they are more accessible to the user than complex models. The training and learning of the model is less time consuming for people who use the simulation model. Similarly, it requires less effort to operate the simulation and to evaluate the results of simple models. Another relevant aspect is that less complex simulations execute in shorter computing time than models with complex rule sets. This aspect results in fast feedback loops and reduced operational costs [NM07].

Complex agents fulfill the presented definition of agency, especially regarding the ability to adapt rules at the individual level. Complex agents are constructed when simple rules

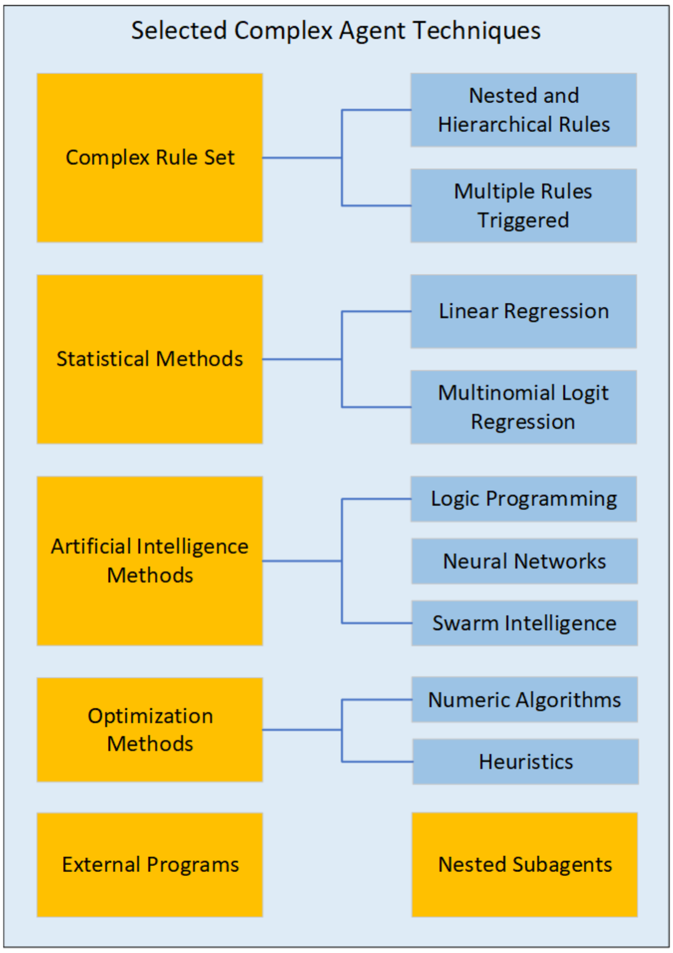


Figure 2.15: Overview of selected techniques for constructing complex agents. Adapted from [NM07].

are not sufficient to model the degree of adaption present in the real-world target. As illustrated in Figure 2.15 there are several approaches to building complex agents.

One approach is to develop agents with a complex rule set, increasing the similarity to the targeted system compared to simple rules. Complex rules share the same structure with simple rules, but have more advanced triggering conditions. Over the course of a single agent decision-making loop, several complex rules may be fired. Additionally, complex rules are often nested and hierarchically structured. When the conditions for multiple rules are satisfied, all of them are triggered. The firing sequence of these selected rules is usually ordered, so that the execution order is not incidental. Nested rules define behaviors that directly influence other behaviors of the agent. Complex, hierarchical rule sets are implemented using either business rule management systems (BRMS), logic programming packages like Prolog[28], or rule engines embedded in general-purpose languages like Java[29]. The application of complex rules has several disadvantages. Compared to simple rules, complex rules require more effort to study underlying patterns, have a more complicated verification and validation phase and result in increased development costs. The most important advantage of complex agents is the enhanced similarity of the model to the target system. This characteristic considerably improves the comprehension of the model and contributes to successful research of the system [NM07].

Some model specifications require even more sophisticated agents and complex rules would be insufficient to adequately simulate the expected agent behavior. There are three different approaches, summarized as *advanced techniques*. The first approach applies statistical methods such as linear regression or multinomial logit modeling. The second technique applies methods from the wide area of artificial intelligence, for example logic programming, neural networks or swarm intelligence. Optimization methods like genetic algorithms and heuristics are briefly discussed as the third option for advanced technique agents.

Statistical methods are used to enhance agents with forecasting abilities to improve the decision-making process [NM07]. Linear regression is a method that finds correlations in quantitative data. In order to find the estimated parameters for the regression line, the ordinary least squares method is used, which minimizes the mean average of the squared errors between the observed dependent variable and the predicted values. Another forecasting method is multinomial logit regression, a generalization of logistic regression. A logistic model is a variation of the regression analysis that models a binary dependent variable. Multinomial logit regression is applied to estimate the probability of discrete events, like acceptance or rejection. This technique enables agents to predict the likelihood that a specific course of action will support their goals or not. Multinomial logit regression assumes that the chance of success is described by the logit function curve. The regression method takes historical data as input to compute the parameters of the curve. The computed function is an estimator of the probability of future events [EKD11] [Win07].

Logic programming is a method in the area of artificial intelligence. In the context of complex agents, it can be used to build a knowledge base that persists and queries complex rules. Rules are stored as asserted facts about the world, which can be recursively compared with queries by an embedded inference engine. An advantage of such knowledge bases is the high expandability as an agent model evolves. Knowledge bases and embedded inference machines are available as ready-for-use software components that can be integrated into agents and bring the potential to reduce development time and costs [NM07] [CF14].

[28] Documentation of the ISO conformant version of the Prolog language (ISO/IEC 13211-1) online at `http://www.deransart.fr/prolog/docs.html` (visited on 03/19/2021)

[29] A comprehensive collection of Java libraries for logical aspects of artificial intelligence and knowledge representation is online at `http://tweetyproject.org/lib/` (visited on 03/19/2021)

Another example from the area of artificial intelligence are neural networks. This approach is applicable for many pattern recognition and prediction problems. For example, these problems involve learning in complex systems with large inputs of various quality, learning in quickly changing systems and pattern recognition in systems with substantial amounts of incorrect data. The swarm intelligence method aims to find a solution to a problem as a group that is more optimal than the single solutions of the members of the group. In agent-based modeling, the swarm intelligence method uses a large number of sub-agents to solve complex problems. In the first step, all sub-agents contribute their parts of the solution to the swarm. In a voting process, agents rate delivered solutions of other agents and one optimal solution is generated. The swarm intelligence technique supplements the nested agent approach presented in this chapter [NM07] [CF14].

Another approach of creating complex agents involves optimization methods, which aim to search for the highest or smallest value among a set of possible values. Optimization methods are either numerical and therefore returning the exact extreme value, or heuristic, returning an approximated extreme value. At a first glance, the prior method seems more advantageous. Such an exhaustive search algorithm would check all possible options in order to determine an extreme value. But while the solution is always optimal, the computation effort might grow exponentially. For the use in agent-based applications, this family of algorithms often is impractical to use and limited to specific problems. The drawbacks of numerical methods resulted in the application of heuristics. Heuristic search algorithms investigate only a restricted set of possible solutions, disregarding unlikely values by calculating an evaluation function. This approach might not be optimal, which means that the best solutions might not be discovered. There is a trade-off between several characteristics such as optimality, completeness or accuracy. Complete solutions contain all extrema and accuracy describes the deviation of the heuristic from the actual optimal solution. Prevalent heuristic algorithms are genetic algorithms, simulated annealing and swarm optimization [NM07].

Another method of building agents is the application of external programs. By integration of agent libraries or external models, existing agent behavior can be reused. This technique results in a reduced implementation effort and less complexity of verification and validation. Agents of complex adaptive systems can be built of entire complex adaptive systems themselves. It is possible to nest agents recursively in a hierarchy of agents. The behavior of high-level agents may be the result of the simple rules of the basic-level agents [NM07].

Agents are the decision-makers in ABMS. While simple proto-agents consist of a set of rules, complex agents feature an advanced toolset such as complex rule sets, statistical methods, artificial intelligence methods, optimization methods, external programs or nested subagents. Agent behavior at the micro-level results in changes at the macro-level, called *emergence*.

2.2.6 Emergence

The core of agent-based simulation models is agent interaction. Agent interaction can be regarded as a process, as it changes in the course of the simulation. Chapter 2.2.1 elaborated that one of the main goals of computer simulation is to achieve an understanding of a certain phenomenon. To understand a complex system, it is beneficial to gain an insight of how the behavioral rules of the agents affect the system. An agent-based simulation model illustrates the process of how the model evolves from a start state to its end state based on agent interaction. The consequence of agent interaction and agent behavior is termed *emergent behavior* or *emergence* [NM07]. Gilbert and Troitzsch state that „Emergence occurs when interactions among objects at one level give rise to different types of objects at another level.“ [GT05] This means that simple rules at the individual agent level may lead to new and complex patterns at the system level. Computational emergence is not only a characteristic of the simulation model, but also a characteristic of the target being investigated [Sym08]. The relationship between the emergent properties of the model and the target is termed *micro-macro link* [Saw03]. Emergent behavior at the macro level is not specified in the model, it is one outcome of the model. Global behavior emerges from the micro level, generated by local agent interactions [MW02].

In the nature, emergence can be observed with a flock of birds. Simple individual bird rules such as *'follow the leading bird at constant speed and angle'* give rise to complex swarm patterns formed by thousands of independent individuals. The imitation effect, defined by simple rules, spreads through agent interaction and forms a communication network. During a simulation, this effect creates a self-reinforcing feedback loop, which builds emergent structures [NM07]. Cellular automata (see Chapter 2.2.3), as predecessors of agent-based models, are suited to discover emergent behavior and to recognize emergent structures. They feature a two-dimensional environment where emergent patterns are visualized in a spatial network [MW02]. Conway's Game of Life simulation model (see Chapter 2.2.3) demonstrates how two simple agent rules result in emergent behavior, generating complex patterns. A similar example for emergent behavior is Thomas Schelling's segregation model of US households discussed in Chapter 2.2.3. Figure 2.14 shows the effects of segregation as emergent pattern. The individuals segregate into areas of the same ethnic group, based on simple rules with local information [Gil02]. There is no central mind that orchestrates the individuals at the system level. Complex agents of ABMS go further than the previously described emergent properties of cellular automata.

While simple agents act solely based on rules, much like ants act based on instincts, complex agents (see Chapter 2.2.5 about agent complexity) have the ability to reason and to decide. The emergent properties that result from such complex agents form human societies. This *second order emergence* distinguishes human societies from animal societies like ant's nests. In human societies there exist not only complex patterns of group behavior. Additionally, the agents themselves are aware of the complex, system-wide emergent patterns and they are capable of adjusting their behavior based on those patterns [GT05].

In the epistemology of computer simulation the type of emergence that has been discussed so far is termed *weak emergence* [Bed11]. The main characteristic of weak emergence is that, in order to study the emergent properties of an agent-based model, the simulation model has to be run and observed. There is no way to otherwise calculate or predict emergent properties based on the initial state of the model. In a system where the actions of individual agents lead to emergent behavior, those interacting individuals remain indepentent agents. This is in contrast to *strong emergence*, where new entities are formed out of the initial agents, like in a chemical reaction. This type of emergence is characterized as irreducible, because it cannot be reduced to the underlying entities [FDP08] [Win19].

2.2.7 Establishing Model Credibility

The certification that a simulation model is valid for a specific purpose is an essential task within the simulation model development process. The stages of the simulation-based research are outlined in Chapter 2.2.2. The set of certification measures mainly consists of the verification and validation processes, which are discussed in this chapter. General verification and validation techniques for computer simulations are presented. Established measures and important steps for agent-based simulations are elaborated in particular. Model logging is outlined as a critical supporting activity for testing agent-based models. Verification and validation are key contributors to simulation credibility. A simulation model that has not been verified and validated, is not ready to support decision-making in business or science.

The goal of the verification measures is to assert that the implemented model works correctly. It must be certified that the model is usable and preferably free of errors, very similar to any other computer software. Correct verification ensures that the implementation operates as the model developers intended. Verification is conducted in order to ascertain that the program code conforms to the design specification, that algorithms are programmed correctly, and that the program does not contain errors, oversights or unintended behavior. In complex agent-based simulations, model results can be unexpected and one challenge is to investigate whether they originate from characteristics of the agents, or from an undetected error in the code [GT00].

Verification does not seek to evaluate the model toward its eligibility or if it resembles the target system appropriately. These issues are the domain of model validation. Validation techniques aim to assert that the simulation is a well-suited model of the target. This means that it correctly addresses the problem statement and that it is able to reproduce the behavior of the target system according to the specification. Validation compares the model against the target, the real-world system the model is intended to resemble [NM07]. Verification and validation are continuous activities throughout the life cycle of model development. This approach aims to identify and correct deficiencies during the development process stage in which they occur [Bal94].

Verification and validation is a complex process. It is complex, because different techniques must be applied correctly and specifically for the model under investigation. It is a process, because there are many steps and iterations involved until the model is validated. The most comprehensive work on verification and validation is published in the book *Verification and validation in scientific computing* (2010) by W. Oberkampf and J. Roy [OR10]. The authors put the antagonism between verification and validation in a nutshell, defining verification as „solving the equations right“ and validation as „solving the right equations“ [OR10]. This definition is a traditional approach, since simulations originate from equation-based systems as discussed in Chapter 2.2.1. The special case of agent-based simulations constitutes additional challenges, especially the verification and validation of agent rules, agent interaction and emergent behavior at different levels [NM07].

Verification

Verification consists of certain actions that are performed to assert that the model meets its specifications from a technical viewpoint. Hence, the main precondition to implement and later verify a model is a requirements and specification documentation of the intended model. Oberkampf and Roy formally define verification as „the process of assessing software correctness and numerical accuracy of the solution to a given mathematical model“ [OR10]. The demanded *software correctness* is a noble goal and an idealized state, but error-free code is only approximately achievable for non-trivial programs. The dutch pioneer in computing science Edsger Dijkstra conceded in 1972 that „Program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence.“ In practice, the developers seek to accomplish a high degree of certainty by applying verification techniques. This certainty is increased in every iteration and with every passed test case. A high coverage of test cases enhances the achieveable level of certainty. In theory, possible test cases could systematically include every permutation of input parameters and branching points in the model. This approach results in a high number of test cases, so it is feasable to automate test runs of the model. Verification describes this iterative process of running tests, identifying program errors and implementing corrections to the code. Specifically, the performed corrections require retesting, so-called *regression tests*, to ensure that no new errors have been introduced [GT05] [NM07].

In practice, verification involves running test cases where the execution of the program code is checked against the model specification document. Verification includes a set of tools that facilitate software testing, such as structured walkthroughs, unit tests and formal methods. These verification techniques, including their strengths and weaknesses, are discussed in this chapter. A prerequisite for applying these methods is the existence of a model documentation. The design documents should be the foundation for the implementation of the model and should be updated regularly to reflect changes. The applied process model could be based on international standards such as IEEE 1012-2016[30], or on a custom project plan. Accompanying design documents range from text

[30] IEEE Standards Association, *IEEE 1012-2016 - IEEE Standard for System, Software, and Hardware*

documents written in prose to structured specifications like UML artifacts. Parallel to the implementation process, the documentation should be updated and expanded to reflect constant discoveries of more detailed model specifications [NM07].

Structured code walkthroughs are manual reviews and static analyses of the program, in which the code is compared to the specification in order to identify errors. This task is usually performed by a group of developers. The authors present their program and manually trace through critical parts of their code line by line, while other developers have the task of auditing the solution for correctness. The objective of code walkthroughs is for the developer to present the solution and to receive feedback if the code is working as intended. It is not an objective that auditing developers propose concrete modifications to the code. This would turn them to contributing authors of the code. While structured code walkthroughs are performed solely based on the source code of the program under investigation, structured debugging walkthroughs actually execute the source code for defined test cases. Debugging in this case describes the task of using a software tool, a debugger, to interactively trace the program code in the order of execution while keeping track of the values of program variables. Modern integrated development environments (IDEs) feature such a debugging tool. By executing the program code, debugging achieves instant confirmation of the existence of fatal errors. Critical values can be examined over the course of defined test scenarios. Given the manual nature of debugging walkthroughs, one main disadvantage is the required time effort. One solution is to automate these tests, so the test coverage can be increased. It is recommended to apply both structured code walkthroughs and structured debugging walkthroughs as complementary techniques for verification of the simulation model [NM07].

Unit testing is a recursive approach to software testing. The basic idea is to run a test for each method, then for each class, for each module and finally test the whole system. A unit test consists of defined input and output values, which are compared to the calculated values. Unit tests assert that the expected solution is generated. Failed tests indicate unintended behavior of the code. With the application of unit tests, there is an established process for software changes. First, the program code is modified and the unit tests are updated to include the modifications. In test-driven-development, the sequence is reverse. Finally, all unit tests are executed to assert the correct operation of the code. Repeated runs of tests are critical to ensure that code modifications do not influence the behavior of other parts of the program, and that no new errors are introduced. The repetitive nature of unit tests suggests the use of automated testing frameworks such as JUnit[31] for the Java programming language [NM07].

The previously presented methods for verification, code walkthroughs and unit tests do not provide any proof of correctness. They cannot guarantee that the code is correct and free of errors. Successfully executed unit tests indicate that for the given test cases, the program is working as intended. But this fact does not allow the conclusion that

Verification and Validation, 2019, https://standards.ieee.org/content/ieee-standards/en/standard/1012-2016.html (visited on 03/19/2021)

[31] The JUnit Team, *JUnit 5*, 2019, https://junit.org/junit5/ (visited on 03/19/2021)

the whole program is correct. There are formal methods to prove that a fragment of code is correct. Formal methods are mathematical and formal logic proofs that use axioms and theorems to prove or refute the problem statement. This approach is mainly used in critical environments, where high demands on software reliability legitimates the additional cost and effort [NM07].

Validation

The task of model validation has been briefly presented in Chapter 2.2.2 as a crucial part of the process of ascertaining that the simulation model is correct and delivers results that correspond to the target, which is an absolute precondition for using the model for analyses or predictions in business or science. While the previously discussed model verification focuses on a correct implementation of the model specification, model validation is concerned about the model specification itself.

Formally, validation can be described as „the process of assessing the physical accuracy of a mathematical model based on comparisons between computational results and experimental data“ [OR10]. Oberkampf further states that „the relationship between the mathematical model and the realworld (experimental data) is the issue“ [OR10]. The goal of validation is to ensure that the model is correctly reproducing and imitating the behaviors of the real-world system.

In the process of using the simulation model, the scientist selects certain cases of the target system and aims to represent them in the model. In further experiments, additional cases, which cannot be studied in the real-world system, can be explored via simulation. This approach raises certain challenges concerning model validation. The acquisition of test or reference cases, for example historical or statistical data, of the target system is often a problem. In some cases, the target system might be completely hypothetic, with no empirical data. Another group of simulation models are not deterministic and feature random processes, which result in different solutions in each run. Finally, the behavior of agents poses certain difficulties in terms of validation. In order to meet these demands and to ensure that simulation models can be applied as scientific tools, certain approaches have been established [NM07].

Similarly to the task of verification, it is not possible to completely validate any non-trivial simulation model in order to assert the perfect reproduction of the real-world system. Hence, the goal is to accomplish a high degree of statistical reliability. This degree of certainty is variable, depending on the target and the specification. Additionally, the level of certainty is hard to measure and it is subjective to the user. In order to increase the level of certainty, more relevant test cases are developed, simulated and examined. Like proofs in mathematics, one validation technique called active nonlinear tests (ANTs), consist of tests to invalidate the model [Mil98]. They include a series of validation and refinement cycles until the resulting model has successfully passed all validation tests [NM07].

In agent-based modeling and simulation, the process of model validation involves the validation of input and output data, the processes that describe the model, and the agent rules and behaviors including the emergent properties. There are several aspects and subtasks concerning the process of validation. The process starts with an evaluation of the specification and the research questions that the model is supposed to answer. The data, derived from the real world system and used in the model, is another candidate for validation. Another validation task examines whether processes in the model correspond to the processes in the real-world. Other items for model validation include model output data and agents. The validation of agents is carried out by analyzing agent behaviors and comparing them to the mechanics of the target system [GT05] [NM07].

One major field of application of agent-based simulation is the modeling of systems that do not support controlled experiments. As a result, it is not possible to collect the data that is necessary to validate the model. The validation procedure of simulation models of physical systems is well-established and documented [Ame98], but in social simulation no such *V&V guidelines* have been developed. While the process of verification can be defined as an objective operation, the model validation however is a subjective task. Still, the goal is to achieve a high level of credibility and plausibility for the model, so that the results are valid with a high statistical confidence. Since agent-based models in social sciences consist of aspects of human decision-making, the model validation process seeks to prove that the model is credible by providing enough evidence like in a legal case. Several approaches for establishing credibility for a model will be presented in the following paragraphs [NM07].

One measure of model validation is to investigate how accurate a model simulates the target system. This validation method centers around covering test cases. A test case consists of predetermined input and output values for a specific situation. Hence, a model is validated when all prepared test cases match the selected cases from the target system. The systematic selection of the most relevant test cases considers observed or inferred cases from the target, commonly presumed situations with a degree of reservation and results from other simulation techniques. When comparing a model to the corresponding target system, three aspects are relevant depending on the availability of observed experiments of the target. Primarily, simulation results are compared to the target system. Alternatively, the results are compared to the results of other, already established models. Another option is the evaluation of the simulation output by experts of that field of research. In practice, simulation models are designed in advance of constructing a physical prototype to investigate implementation details and expected behavior. In this case, no data is available for comparison to the simulation results and alternative test cases have to be devised [NM07].

Another validation approach requires real-world test cases as foundation for estimating model parameter values. This method is called model calibration. A subset of the test cases is used for model calibration, while the rest of the test cases are utilized for validation. One technique for identifying possible test cases is parameter sweeping. This technique explores the range of all potential input parameters to the simulation model.

By eploring the whole variety of possible results and behaviors, the most relevant cases are identified and selected for deeper analysis. Systematic variation of input parameters is classified as sensitity analysis and parameter sweeps, which will be discussed later in this chapter [NM07].

The range of model validation approaches is further increased by the method of using multiple models to validate each other. This approach seems redundant and a suboptimal allocation of resources at first glance. But the work of Hales at al. [HRE03] showed that comparisons of two or more models are a useful tool for model validation. When applying different simulation models to the same problem, the models exhibit differences regarding their modeling of the real world, but they should agree on the results. Two agreeing models do not necessarily validate each other, but add another argument to the model's credibility. In the case of disagreeing models, the validation task provides references for deeper analysis of the underlying reasons for the different results. Corresponding to other scientific work, it is feasable to independently reiterate simulations with different models in order to validate the theory [NM07].

The next validation method to be discussed regards agent-based models as a special case of an analytical, or noncomputational, model. It is possible to reduce an agent-based model to an analytical model, expressed by equations, where the results can be computed directly. Then this analytical model as simplified version of the agent-based model should compute the same results as the computational model. If the analytical model is proven to be correct by a verification and validation process, it can be used as validation tool for the ABMS model [NM07].

Subject matter experts can be a relevant contribution to the process of model validation. They have the ability to offer detailed exptertise and have the knowledge to select relevant test cases fo model validation [NM07].

The validation of agent theories is an ongoing challenge concerning the validation of simulation models. Precisely, agent theories include agent behaviors, agent interaction and emergent features that result from the model. At present, there is no generally applicable theory about human behavior that can be applied to any agent-based model. Programs in the field of artificial intelligence have not yet delivered practical social models. Yet, there are niches where social interaction is technically well described, for example the field of supply chain management [MSN04]. The translation of social interaction theories into a technical agent-based model is a field of research in the area of computational social simulation. This thesis aims to contribute to this field of research regarding the routine activity theory by Marcus Felson and Lawrence E. Cohen [CF79] (see Chapter 2.3) and its implementation in an agent-based simulation model.

Model Logging

A precondition for verification tests is the collection of model output data. Logging simulation results is especially significant for agent-based simulations, because they tend to generate very detailed information at different levels. This multidimensional data originates from the system level and from the agent level, generated by agent behavior, agent interaction and agent decisions. The generated data should be saved in a structured form to facilitate efficient model testing and analysis. Usually, the model output log is structured as time-series data, representing the states of agents, possibly the states of groups of agents and the system states over time. Time-series data is a structure which assigns a time value to each value of the logged attributes. Thus, the output is a sequence of discrete-time data. At specified discrete time intervals such as ticks, minutes or years, the simulation records the value of each variable of interest together with the according time stamp. Agent-based models have the potential to produce very large logs, which allow the researcher to analyze the model in great detail. But the accuracy comes at a cost. In Chapter 2.2.2 there is a calculation of a sample log file size for a medium-sized simulation of 1000 agents and their interactions over one simulated year. The result is a simulation log of 42 048 Gigabytes of data, which poses challenges in terms of storage space and processing speed for further analysis. One important measure to control the volume of simulation output data is aggregation. Aggregate results are summaries of agent attributes or system states. These results are computed from time series data of agent behavior and contribute to analyze and understand model results [NM07].

Generating model logs at an increasing level of detail also affects the execution speed of the simulation as well as the required time and effort for examining the results. Researchers counteract this dilemma by prioritizing simulation output data. This approach involves identifying the relevant attributes for the output log and deciding which attributes to omit. This concept is often supported by the simulation package with customizable levels of logging. It allows researchers to deliberate about whether generating more data at the expense of storage space and execution speed is beneficial for the goal of the study. Three common approaches to agent-based simulation logging are examined [NM07].

The first approach enables the researcher to log data by content. This topical approach includes both data at the agent and at the system level. Possible attributes are agent events such as robberies, prevented robberies, agent statistics such as time spent at home, agent groups such as citizens or system-wide properties such as the emergence of hot spots. Please refer to Chapter 6 for an in-depth discussion about the logged results of the simulation research based on this thesis. While topical logging enables researchers to customize the output data based on certain attributes, it does not allow to specify the granularity within a specified attribute.

This leads to the second approach, which classifies log output by level of detail. Following this idea, the users choose the granularity at which attributes are recorded. Possible output options with increasing level of detail are *log errors*, *log events*, or *log all data for debugging*. Depending on the stage of simulation-based research, the user is able to select the most suitable output, because demands on output logs vary from implementation phase, to verification of the model and analysis of the results.

The third approach combines the previously discussed logging strategies in order to accomplish improved flexibility. This method allows users to independently select the logging level of detail for each topic. While this approach is the superior logging method, it requires additional development effort and sophistication. This method is recommended for large-scale or complicated simulation models [NM07].

Sensitivity Analysis and Parameter Sweeps

Oberkampf defines sensitivity analysis as „the determination of how a change in any aspect of the model changes any predicted response of the model“ [OR10]. In a simple approach, one parameter is changed around kay values, producing a change in the output. This method exhibits three limitations. First, the dependence between input and output parameters is not always linear. Second, other parameters may be influenced. Finally, the results depend on the selected units. One approach is to compute normalized sensitivity by calculating the proportions between the changes. Another approach is to introduce uncertainty by describing a change by its standard deviation [Nor15].

A common method of testing the sensitivity of a simulation is to introduce randomness to the model, thus computing a distribution of results. This way, the parameters are drawn from a uniform random distribution. Investigating the output of the simulation generated from a series of runs will provide indications of the relationship between the variables and the results. Random numbers are also used to account for external processes, or exogenous factors, that are not part of the model. In this case, the random value substitutes unmeasured or unknown parameters in absence of precise information. In order to draw conclusions from a sensitivity analysis, the simulation must be executed repeatedly and the results must be prepared as distributions [GT05]. Other, more sophisticated methods for sensitivity analysis include fourier amplitute sensitivity testing, regionalized sensitivity assessment and sampling-based sensitivity analysis [Nor15].

In agent-based modeling, sensitivity analysis is further used for investigating agent rules. By modifying the behavior of an agent, the effects on the system can be tested. Sensitivity analysis can be applied for balancing the agent rules between simple and complex in order to develop the most basic set of rules which still reproduces the behavior of the target. If a model is too simple, it lacks credibility and if it is too complex, it is difficult to obtain information of agent behavior [NM07].

A more radical approach to sensitivity analysis is *parameter sweeps*, where one or more variables are systematically varied in defined increments within a specified range to identify all possible behaviors of the model and their corresponding conditions. This range of parameters may vary depending on the subject from large to small steps. For example one parameter p defined as $p \in [0.1, 1]$ is tested using ten values starting from 0.1 with increments of 0.1. A simulation run is conducted for each of the ten values, exploring the full range of the parameter space of the parameter p. This process is repeated for all variables of interest, resulting in a full range of possible model results. In order to automate this repetitive task, many simulation packages have built-in features for this task. Computational demands naturally set limitations to parameter sweeps, so a large count of variables of interest and broad value ranges lead to larger intervals and less coverage for the examined variables. The gap between the intervals can be adjusted in cyclic phases [NM07].

One variation of the parameter sweeps method effectively reduces the number of required model runs. This method applies stochastic search algorithms to randomly choose the next step within the parameter space. One downside of this method is that more relevant values have the same chance of being chosen by random sweeps as uninteresting values. Randomness adds another approach to exploring the range of model results during the process of parameter sweeps. In most agent-based models, randomness is an inherent feature. It is used to account for uncertainty in agent behavior and environment processes. Randomness is modeled by probability distributions for those uncertain parameters. As part of the parameter sweeping process, the model is repeatedly run with varying random seeds for the same set of input parameters. This stochastic method adds more options to the model certification process at the expense of additional computational resources [NM07].

2.2.8 Simulation Software for Agent-Based Modeling

Agent-based modeling and simulation software is used in various settings, ranging from home desktop applications, office or enterprise environments, to scientific large-scale ABMS. In this chapter, software solutions for agent-based models are discussed based on their increasing demands on personnel and hardware investment. All approaches have their strengths and weaknesses. One ABMS package might be the optimal solution for a specific range of problems and a sub-par tool for others.

Architectures of ABMS Software

Common ABMS architectures feature a three-tiered approach. The simulation engine computes the simulation tasks. The interface enables users to manipulate variables, to control the simulation and to read output data. The data storage provides a means to save settings, agents and environments. These three architectural layers formulate core functions, commonly implemented in modules. Many agent-based software packages are comprised of all three core functions, bringing all necessary features for rapid development. Other tools focus on one specific core function and enable users to choose modules for other functions according to their requirements. The three-tiered ABMS architecture is commonly implemented in tightly coupled, loosely coupled or distributed architecture styles [NM07].

The first group unites the simulation engine and the user interface in a single program. Many desktop ABMS such as NetLogo or MATLAB can be regarded as tightly coupled architectures. This group of programs feature reduced complexity at the cost of scalability and flexibility. Tightly coupled architectures are built of fewer modules than other styles, which simplifies communication between modules, for example direct function calls instead of a multi-layered API. Tightly coupled architectures are well suited for small-scale ABMS up to several thousand agents, depending on agent complexity and the amount of data used [NM07].

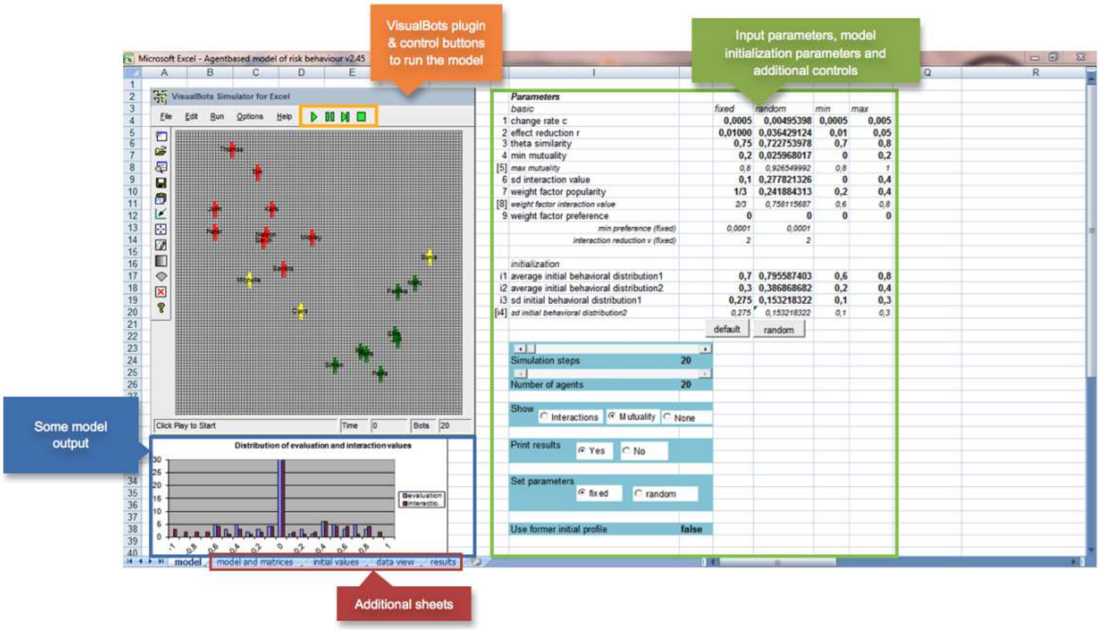
Loosely coupled architectures separate the simulation engine and the user interface into independent modules. A possible example is a C++ program with a JavaScript web interface. Loosely coupled architectures feature more modules and exhibit an increased complexity. Due to the separation of modules, the communication between modules is more challenging. In contrast to distributed architectures, the simulation program is usually implemented in one single module. Being a balanced trade-off between complexity and scalability, loosely coupled architectures are applied for simulations of medium size, up to hundreds of thousands agents [NM07].

Distributed architectures further separate the simulation engine and the user interface into multiple processes on different computers. This approach allows the best scalability for high-performance computing, at the cost of increased development complexity and resource costs. Distributed architectures are built for large-scale ABMS with possibly millions of agents [NM07].

Desktop ABMS

A standard personal computer is sufficient for performing agent-based modeling and simulation on the desktop. Simulation software for desktop environments are usually designed in a tightly coupled architecture and they are used for small-scale simulations, learning, prototyping and for limited analysis. Agent-based simulation programs for the desktop can be classified in spreadsheets, computational mathematics systems and dedicated simulation environments. Spreadsheets are common tools in home and business environments. It is possible to build basic agent-based simulations in spreadsheets with scripting capabilities. The second group, computational mathematics systems, provide advanced modeling and mathematic problem solving libraries, as well as more sophisticated output capabilities than spreadsheet programs. Being general-purpose programs, they often lack specific capabilities commonly used in simulations. Finally, agent-based simulation packages, or prototyping environments, deliver all required components for modeling and simulation, including specific features for agent handling, output, or visualization [NM07].

Any spreadsheet program that supports multiple worksheets and a scripting language, for example Microsoft Excel, LibreOffice Calc or Google Sheets, can be used to implement small-scale ABMS. A worksheet consists of a set of rows and columns, which can be addressed by their X and Y locations. A single cell is referred to by its row and column and contains text, a number or a formula. In order to extend the functionality of supplied functions, it is possible to enter user-defined functions and subroutines. These functions serve as the simulation engine. Model input parameters are defined in cells and the model output can be visualized in tables and charts. Figure 2.16 shows an example for a small-scale agent-based simulation implemented in Microsoft Excel.



VisualBots plugin & control buttons to run the model

Input parameters, model initialization parameters and additional controls

Some model output

Additional sheets

Microsoft Excel - Agentbased model of risk behaviour v2.45
VisualBots Simulator for Excel
File
Edit
Run
Options
Help
Click Play to Start
Time
0
Bots
20
Distribution of evaluation and interaction values
model
model and matrices
initial values
data view
results

	Parameters				
	basic	*fixed*	*random*	*min*	*max*
1	change rate c	0,0005	0,00495398	0,0005	0,005
2	effect reduction r	0,01000	0,036429124	0,01	0,05
3	theta similarity	0,75	0,722753978	0,7	0,8
4	min mutuality	0,2	0,025968017	0	0,2
[5]	*max mutuality*	*0,8*	*0,926549992*	*0,8*	*1*
6	sd interaction value	0,1	0,277821326	0	0,4
7	weight factor popularity	1/3	0,241884313	0,2	0,4
[8]	*weight factor interaction value*	*2/3*	*0,758115687*	*0,6*	*0,8*
9	weight factor preference	0	0	0	0
	min preference (fixed)	*0,0001*	*0,0001*		
	interaction reduction v (fixed)	*2*	*2*		
	initialization				
i1	average initial behavioral distribution1	0,7	0,795587403	0,6	0,8
i2	average initial behavioral distribution2	0,3	0,386868682	0,2	0,4
i3	sd initial behavioral distribution1	0,275	0,153218322	0,1	0,3
[i4]	*sd initial behavioral distribution2*	*0,275*	*0,153218322*	*0,1*	*0,3*

default random

Simulation steps 20

Number of agents 20

Show ○ Interactions ● Mutuality ○ None

Print results ● Yes ○ No

Set parameters ● fixed ○ random

Use former initial profile false

Figure 2.16: An agent-based model of risk behavior in a spreadsheet [SBG14].

Computational mathematics systems are an established alternative to spreadsheets and multi-agent programming environments. They offer extended scalability, but can be difficult to learn and adapt. Most general-purpose computational mathematics systems can be used for ABMS. The essential feature is the integrated scripting language. One advantage of computational mathematics systems is their rich arsenal of mathematical functions. Some programs support symbolic variables for solving systems of equations. Symbolic variables are variables without values assigned to them. The strengths of computational mathematics systems lie in small to medium-sized agent-based models with mathematically focused processes and rules. The integrated development environment (IDE) of modern computational mathematics systems support implementation, debugging and documentation of the simulation program. Another advantage is their set of output facilities, including charts, graphs and other visual representations of data. Some large-scale agent-based modeling and simulation toolkits facilitate the output capabilities of mathematical environments for presenting their simulation results. The main disadvantage of applying computational mathematics systems for agent-based modeling and simulation is the absent agent functionality. Agents, their attributes, rules and behaviors have to be programmed by the user. [NM07].

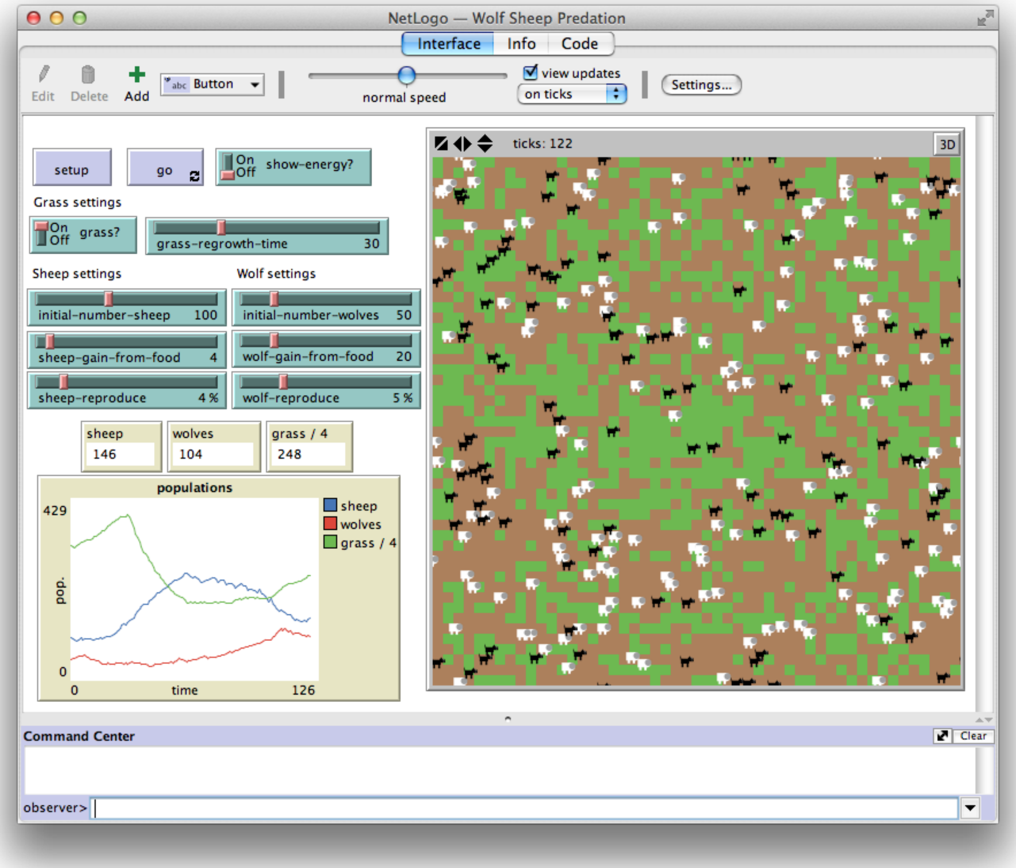
MATLAB and Mathematica are two representatives of computational mathematics systems that can be used for agent-based modeling and simulation. MATLAB (former *matrix laboratory*) is a numeric computation system. Originally developed at the University of New Mexico in the 1970s for performing numerical linear algebra, it is now developed and commercially distributed by MathWorks. MATLAB is used in education, science and industry. The program's features include an object-oriented programming language, operations on arrays and matrices, and 2D/3D plots for data visualization. Third party programs written in C, FORTRAN or Java can communicate with MATLAB using its API [NM07] [The19b].

Mathematica is a commercial numeric computation system and a widely used program in mathematics and nature science. Mathematica contains a programming language which unites elements of object-oriented, functional, rule-based and procedural paradigms. It supports symbolic processing of equations as well as numerical solving of equation systems. A built-in toolkit for visualization facilitates 2D and 3D graphics. In 2006, an alternative user interface for Mathematica was introduced. Wolfram Workbench is an IDE based on the Eclipse project [Gui] and provides development tools including debugging, testing and revision management [Wol19].

There is a wide range of commercial and open-source agent-based modeling platforms available [NM09]. After a general introduction, a selection of significant software products is discussed. While the ABMS programs NetLogo and AnyLogic are briefly presented in this chapter, the Repast toolkit is discussed in greater detail separately, because the simulation model of this thesis is implemented using the Repast-based tool Agent Analyst.

Dedicated agent-based modeling and simulation environments are not general-purpose calculation programs like the previously presented spreadsheets and computational mathematics systems. They are programs specifically built for agent-based modeling and simulation. This specialization allows them to provide built-in support for agent development. The selected simulation platforms for desktop environments share certain characteristics. The tools are focused on a visual user interface. Their target audience include first-time users and students as well as professionals in industry and science. Toolboxes for basic modeling components, for example schedulers, variable trackers or output facilities, are built-in and ready for use. One common component of agent-based modeling platforms is built-in support for agent development, allowing users to manage attributes, rules and behavior at the agent level. In order to support less experienced modelers and save development effort, common agent functions and generic agent capabilities are provided. The reduced complexity is achieved by the automation of common modeling and simulation tasks. These features for agent development support require less training and result in reduced development effort, but also limit their flexibility. Compared to computational mathematics systems, dedicated desktop simulation programs are limited in terms of mathematical libraries and output capabilities [NM07].

NetLogo is a dedicated ABMS environment and programming language. It is considered a tightly coupled architecture. NetLogo was created by Uri Wilensky at the Northwestern University Center for Connected Learning and Computer-Based Modeling for educational purposes and it is freely available under a GNU General Public License. It uses a derivative of the Logo programming language and provides a graphical user interface with an IDE [NM07]. The central concepts of NetLogo are *turtle* agents, living in an environment of *patches*, controlled by the *observer*. While NetLogo is predestined for



NetLogo — Wolf Sheep Predation

ticks: 122

grass-regrowth-time 30

initial-number-sheep 100

sheep-gain-from-food 4

sheep-reproduce 4 %

initial-number-wolves 50

wolf-gain-from-food 20

wolf-reproduce 5 %

sheep 146

wolves 104

grass / 4 248

☑ view updates

on ticks

normal speed

observer>

Figure 2.17: Screenshot of the desktop ABMS program Netlogo 6.0.4 (2018) [Wil99b].

life models, it provides an extensive model library from the domains of biology, physics, economics and system dynamics [Wil99b]. A screenshot of the graphical user interface of NetLogoo is shown in Figure 2.17. It is an implementation of the „Wolf Sheep Predation Model“, investigating the stability of predator-prey ecosystems [Wil99a].

AnyLogic is a multi-platform commercial modeling and simulation environment. It currently features three licensing models, including one free version for beginners and students. AnyLogic is developed by The AnyLogic Company and it is based on the Java technology and on the Eclipse Framework. AnyLogic supports three modeling methodologies: agent-based modeling, discrete event and system dynamics. It is possible to create models using a combination of these modeling methodologies. AnyLogic features a built-in integration with GIS maps by supporting the shapefile standard SHP by ESRI (see Chapter 2.1.4), and tile maps like OpenStreetMap. By using an imported map as environment for the simulation model, the user is able to access data from maps and to define geospatial routes for agents. Other libraries provide common routines in the areas of process modeling, pedestrian flows, rail yards, fluids, traffic and production. Models in AnyLogic are implemented by using a graphical modeling language. Model elements in AnyLogic can be extended using Java code or third-party Java libraries. The tight integration with Java enables users to build small to medium-scale, complex simulation models. One disadvantage is that internal procedures like event-handling or the simulation engine are hidden from the user [Ste12] [The19a].

Large-Scale ABMS

Large-scale agent-based modeling and simulation expands the previously discussed desktop programs to servers and computer clusters. The increased and scalable computing power facilitates a higher number of interacting agents, more sophisticated agent rules and behaviors and more complex models. Large-scale ABMS programs also run on desktop systems, but due to their advanced feature set, they require more engineering resources and know-how. The main characteristic of large-scale ABMS is their loosely coupled architecture and the resulting high degree of modularity. Among their other distinguishing concepts are advanced scheduling, the addition of local time and agent state management [NM07].

In agent-based simulations, the *scheduler* manages the passing of time. Agents require a scheduler for performing their actions relative to local or global time. The scheduler determines when agent actions are performed and in which order they are performed. There are two techniques for time scheduling in agent-based simulations. Time-step scheduling is a simple, but inflexible approach. Descrete-event scheduling is more sophisticated, but allows more detailed event sequences and extensible models [NM07].

Time-step scheduling is based on an integer clock to model the flow of time. Time steps start from zero and are subsequently incremented by one until the simulation is stopped. These time steps from the set of positive integers correspond to a time unit defined by the analyst. Every agent action occurs at one of these time steps. For example, if the model is implemented with one time step corresponding to one minute, then the 1440th tick marks the passing of one day and one year is reached after 525 600 ticks. Time-step scheduling is created to support multiple actions from many agents at the same time step to simulate parallel events. This raises the problem of the execution order of agent actions. The scheduler is responsible for deciding this order. In a simple approach, the actions are executed in the order they arrive at the scheduler. This approach has the disadvantage that an arbitrary ranking of agents is established, where the leading agents always perform their actions first. Very often this behavior is not the intention of the modeler. As a result, more sophisticated schedulers support rotation and randomization of agent actions. Time-step scheduling owns the immanent drawback of limited flexibility. For example, the refactoring of a model from a daily to an hourly level requires rewriting of all bahaviors. In order to enhance the flexibility of models, modelers either choose a time unit that is finer than the current behavior dictates, or employ a discrete-event scheduler [NM07].

Discrete-event scheduling replaces the integer clock with a floating point number of single or double precision. By this modification, events are still allowed to occur at steps 1, 7, or 525 600, but also at 1.75, 7.8 and 525 600.349 872. Events are still discrete, because they occur sequentially. It is possible though that locale schedulers introduce parallel execution of events. Every event has a time stamp assigned to it, which determines the order of execution. In the case of identical time stamps, the same techniques as for time-step scheduling are applied: rotation and randomization. Discrete-event scheduling enhances the flexibility of the model. For example, each time step represents one day. Then it is possible to introduce a new time scale like hours, corresponding to a rounded 0.041 667 of one step, without the necessity of rewriting agent behavior code [NM07].

The simulation scheduler also manages global time, while optional local schedulers manage local time. This approach enhances the modularity of the model. Now it is possible to model the flow of time within subsets of agents, effectively hiding details of the micro-level from the macro system. Another application of the local time concept is to perform as an adapter for legacy models implemented in a different time scale. The major benefit of local time schedulers is the scalability with the number of processors. In agent-based simulations, events are often independent from other events. The scheduler stores future events waiting to be fired in a queue until the current events are finished. The concept of bounded rationality (see Chapter 2.2.4 for a definition) allows the parallelization of events, orchestrated by the scheduler. Large-scale agent-based simulations that run on multiple processors in computer clusters directly benefit from concurrency and run the simulation more efficiently [NM07].

Another main feature of large-scale agent-based modeling and simulation tools is a facility to persist simulation states at any time step. State saving includes both agent and environment attributes and enables analysts to view and investigate them outside the simulation, for example in spreadsheet applications. Additionally, state saving is used to start the simulation from a defined starting point. This procedure named *hot starting* allows to skip an initial warm-up period, during which the simulation reaches the desired state. The method is especially useful for repeated experiments, sensitivity analysis and tests that share the same initial state [NM07].

Two notable representatives of large-scale agent-based simulation tools that incorporate the previously presented set of features are Swarm and Repast for High Performance Computing (Repast HPC). Swarm is an open-source ABMS package initially released at the Santa Fe Institute in 1997 and maintained by the Swarm Development Group. There are scientific and commercial applications of Swarm in the domains of investigating emergent behaviors in biological systems, supply chain optimization and others [Min+96] [NM07]. Repast for High Performance Computing [CN13] is part of the Repast suite that is comprehensively discussed in the following chapter. Repast HPC is an agent-based simulation toolkit specifically intended for distributed computing. It is based on Repast Simphony and written in C++ to work in parallel distributed systems. Repast HPC has been successfully applied in supercomputing projects, including Argonne National Laboratory's Mira and Theta computing facilities [Arg18].

The Repast Landscape

The Recursive Porous Agent Simulation Toolkit (Repast) is a widely used open-source agent-based modeling and simulation software. Repast supports agent models with a focus on social interaction. Repast was originally developed in 2000 by Sallach, Collier, and others at the University of Chicago and it is currently maintained at the Argonne National Laboratory. All versions of repast are open-source under the „New BSD“ license. The Repast landscape consists of several programs. Repast for Python Scripting (Repast Py), Repast for Java and Repast.NET are discontinued, while Repast Simphony [Nor+07] and Repast HPC [CN13] are the current maintained versions [Nor+13].

Repast J, the original implementation, and Repast Py share the same features, except that Repast Py provides a graphical user interface for rapid agent development. It uses a derivative of the Python programming language (Not Quite Python, NQPy) to implement agent actions. NQPy allows seamless integration of Java constructs within the Python code, enabling users to export their Repast Py models to Repast J. Repast Py is targeted at small-scale models and rapid prototyping while Repast Java targets more experienced users building large-scale models [Joh13]. The Repast suite is designed in a loosely coupled architecture and can be considered a large-scale ABMS with a field of application from desktop to computer cluster environments.

The functionality of Repast is organized in optional and core modules. Users are enabled to add Java code and Java libraries to extend those modules. The six core modules are the Engine, the Interactive and Batch Run Modules, the Adaptive Behaviors Module, the Domains Modules and the Logging Module. The components of Repast are shown in Figure 2.18. The Engine performs all activites in a simulation and it contains the Controller, Action, Scheduler and Agent parts. The Controller is responsible for starting, pausing, stopping and incrementing the simulation. Actions are events induced by agents during a simulation. Schedulers model the flow of time in discrete event steps and fire agent actions. The Logging Module brings features for managing simulation output. It consists of two modules, the Data Logger and the Object Logger. The Data Logger is a basic facility for persisting primitive values such as integers, strings or booleans. Object Loggers are more complex and have the ability to persist the complete state of an object, for example an agent. The Interactive and Batch Run Modules are the link between the Engine and the user. They contain user interface elements for controlling the simulation, visualizations for a graphical output of the simulation environment, data graphs and probes. Probes support direct manipulation of the model and agent variables. The Adaptive Behaviors Module provides methods for complex agent behavior, including techniques from the domains of neural networks, genetic algorithms and learning models. The Domains Module incorporates genre-specific functions, for example from social sciences, GIS, system dynamics or game theory [NCV06].

Repast Simphony is the successor of Repast 3 with a stronger emphasis on a modular plug-in architecture. Each plug-in is designed in layers, allowing different implementations of low-level functionality within plug-ins. Repast Simphony is backward compatible

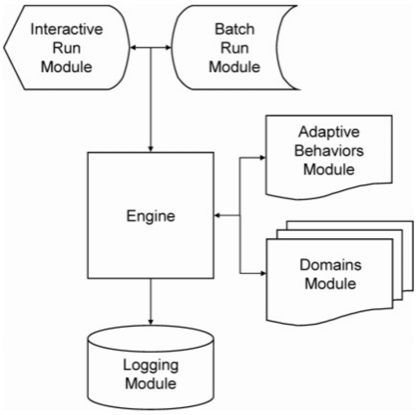


Figure 2.18: The components of the Repast J Agent Modeling Toolkit [NCV06].

with Repast 3 models. It also utilizes Eclipse as IDE, providing different views and perspectives for developing Repast model components like ReLogo projects, flowcharts and agents. Figure 2.19 shows the workspace of Repast Simphony and the zombie demonstration model. Repast Simphony supports different approaches for developing agent-based models.

The first technique is the ReLogo language, an agent-based modeling language for creating models in the open-source Repast suite [Ozi+13]. ReLogo combines the Repast ABM functionality with the Logo programming language, similar to the previously presented NetLogo ABM software. Alternatively, users are able to implement models in the Java or Groovy language. Groovy compiles to Java bytecode and it operates seamlessly with Java code and libraries. ReLogo developers may write Groovy or Java code within their ReLogo source file to utilize the strengths of the three programming languages. Another agent-based modeling method supported by Repast is statechart modeling. Statecharts provide a visual representation of states of components and transitions from one state to another. Statecharts represent complex agent states and behaviors and they can be used to monitor agent states during the simulation [Ozi+15].

Repast Simphony can be regarded as a collection of plug-ins, enabling users to customize Repast to meet their modeling demands. The Simphony Application Framework is the central plug-in and provides the modular user interface and the plug-in system. The Repast Core plug-in set offers general simulation methods, like the discrete event time scheduler, behavior management and random number generation. The GIS plugin provides functions to correlate agents to geospatial data. Agents can be represented in space as point, line or polygon. Repast Simphony utilizes GeoTools, an open-source Java code library maintained by the Open Source Geospatial Foundation (OSGeo). Repast Simphony is compatible with the ESRI shapefile data format (see Chapter 2.1.4). The Freeze Dryer plug-in provides functionality to persist the state of a simulation at any step for later import. The objects are serialized into a table or alternatively to XML.

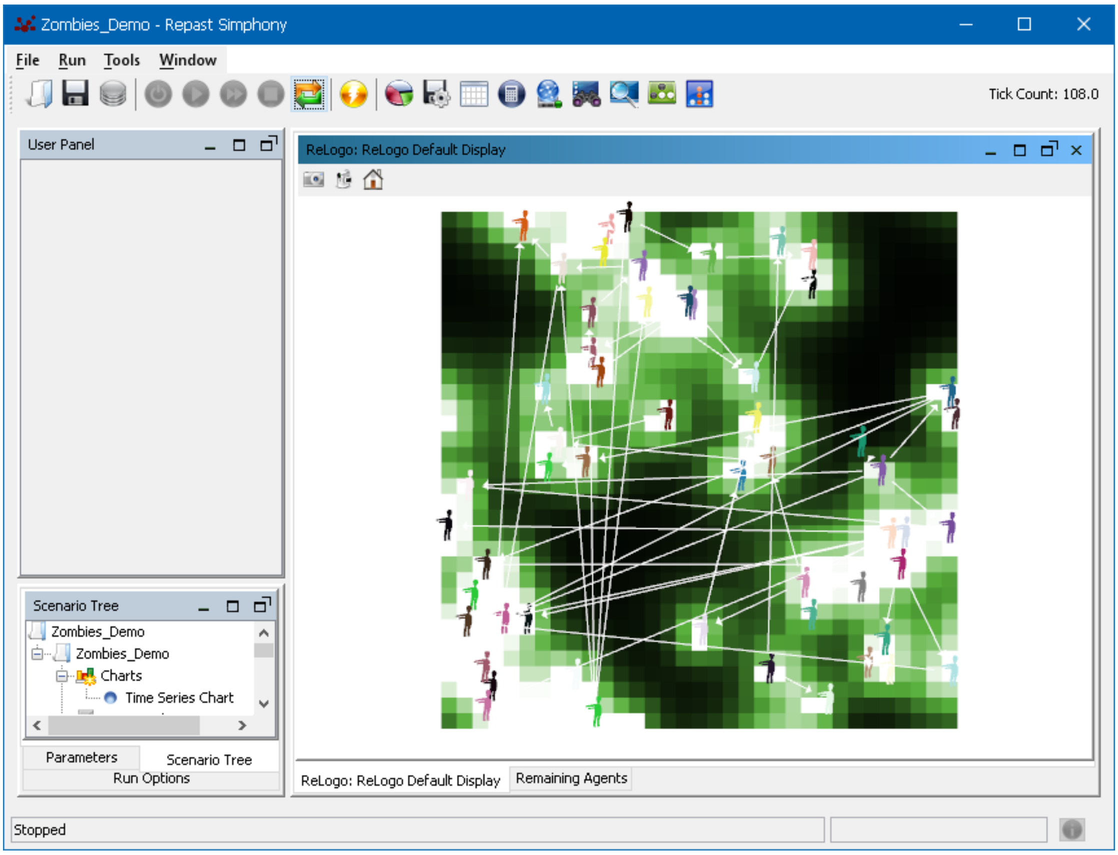


Figure 2.19: Screenshot of the open-source agent-based modeling and simulation platform Repast Simphony 2.6 (2018) [Arg18].

This allows users to create snapshots of simulation states and import those states into common tools like Microsoft Excel. The Repast Simphony package contains several other plug-ins, like modules for batch execution, 2D/3D visualization, distributed model execution, charts and a range of third-party applications including the R environment for statistic computing[32], GRASS GIS (see Chapter 2.1.6), *ORA network analysis system[33], JasperReports[34], and the Weka data mining system[35] [Nor+13].

[32] The R Foundation, *The R Project for Statistical Computing*, 2019, https://www.r-project.org/ (visited on 03/19/2021)

[33] CASOS, Carnegie Mellon University, **ORA* [Car14], 2019, http://www.casos.cs.cmu.edu/projects/ora/ (visited on 03/19/2021)

[34] Jaspersoft Corporation, *JasperReports Open Source Java Reporting Library*, 2019, https://community.jaspersoft.com/project/jasperreports-library (visited on 03/19/2021)

[35] University of Waikato, *Waikato Environment for Knowledge Analysis (Weka)*, 2019, https://www.cs.waikato.ac.nz/~ml/weka/index.html (visited on 03/19/2021)

Method	Number of levels	Communication between agents	Complexity of agents	Number of agents
System dynamics	1	No	Low	1
Microsimulation	2	No	High	Many
Cellular automata	2	Yes	Low	Many
Learning models	$\geqslant 2$	Optional	High	Many
Agent-based models	$\geqslant 2$	Yes	High	Many

Table 2.7: Comparison of the discussed simulation methods. Adapted from [GT05].

2.2.9 Summary

This chapter provided an introduction to agent-based social simulation. Various methods and approaches for computer simulation were presented. Table 2.7 shows a comparison of the discussed simulation methods and lists their characteristics. Methods that allow communication between agents, support complex agents and a high number of agents are best suited for modeling human behavior. A special emphasis was made on agent-based modeling and simulation. Several aspects of agents were discussed in detail. The properties of agents include the concepts of autonomy, social ability, reactivity, proactivity and adaptivity. Social behavior was characterized as interaction between agents. The distinguishing attributes of proto-agents and complex agents were established and techniques for constructing complex agents were presented. The phenomenon of emergence was introduced as the consequence of agent interaction. According to the presented stages of simulation-based research, the processes of verification, validation and sensitivity analysis have the goal to establish model validity and credibility. The chapter concludes with an overview of simulation software.

2.3 Routine Activity Theory

The routine activity theory was developed by the sociologists Lawrence E. Cohen and Marcus Felson from the University of Illinois, Urbana, and published in a paper titled „Social Change and Crime Rate Trends: A Routine Activity Approach“ [CF79]. The work focuses on the circumstances of crime. The authors explained crime as the convergence of likely offenders, suitable targets and the absence of capable guardians. The motivated offender may belong to the group of unemployed teenagers or to the addict population. The suitable targets mainly include people carrying easily transportable goods with a high value such as cell phones, wallets, cameras or jewellery. The guardianship element refers to present law enforcement officers, civil bystanders or security cameras. The convergence of these three elements must occur in space and in time to allow the occurrence of crime. Cohen and Felson further stated that social structures have an impact on this convergence and as a result on the frequency of crime. The key concept of the paper is that „the dispersion of activities away from households and families increases the opportunity for crime“ [CF79]. The goal of this work is to evaluate this concept based on an agent-based simulation model.

Street robbery is a violent crime and it is also classified as *instrumental crime*, which is about economic gain and therefore a result of rational choice rather than an expressive crime. The FBI defines robbery as „the taking or attempting to take anything of value from the care, custody or control of a person or persons by force or threat of force or violence and/or by putting the victim in fear“ [Cri19]. Street robberies, as a type of robbery, are most common in cities and happen in open areas.

The routine activity theory belongs to the field of crime opportunity theory, which is related to the rational choice theory. The rational choice theory is concerned about the causation of crime. It regards humans as rational entities who evaluate the possible gains and losses in order to make a rational decision. The rational choice theory assumes that humans who commit crime carefully plan their actions and think that the benefits of a criminal action outweigh the possible penalties. Offenders select their targets deliberately and also choose the place of the crime with purpose [Sie10].

2.3.1 Summary

Lawrence E. Cohen and Marcus Felson investigated a sociological paradox: While the social and economic situation in the United States improved between 1960 and 1975, reported rates of violent and non-violent crime increased. In 1979, Cohen and Felson postulated the *routine activity theory*, explaining that the paradoxical crime rate changes relate to a shift in the routine activities of citizens away from their homes. These changes especially affect a class of crime known as street robbery. Cohen and Felson further elaborated that robbery requires three elements to occur: a motivated offender, a suitable target and the absence of a guardian. These three elements must converge in space and time for the offense to happen. Cohen and Felson argued that an increase in convergences leads to an increase in the number of criminal acts, even if the number of potential offenders and victims remain constant [CF79].

Cohen and Felson showed that social structures have an impact on spatial patterns of crime. People live in suburbs and commute to their workplaces in the city. This movement leads to an increase in convergences. The criminal propensity as the subgroup of the population, which is inclined to commit crime, is not investiged but assumed given. As previously stated, the minimal elements of the street robbery crime type are an offender with criminal propensity, a suitable target and the absence of guardians. The lack of one of them prevents the criminal act from occuring. The concept of guardianship implies not only the presence of law enforcement officers preventing the criminal act, but also the presence of other civilians that successfully prevent the criminal act from occuring. The result of present bystanders could be that the motivated offender decides not to rob the suitable target because there would be witnesses or bystanders, which leads to a higher chance of being caught and punished for the offense. The frequency of crime, or number of criminal acts per unit of time, is affected by these three minimal requirements [CF79].

Cohen and Felson further investigated the ecological nature of crime. The relationship between offender and victim can be referred to as predatory. From a macroeconomic perspective, predatory actions such as street robbery do not increase the wealth of the society, but only change the distribution of wealth. Criminal acts depend on legal activities, so the spatio-temporal structure of routine activities plays a role in determining the frequency and the location of criminal acts. Daily activities cause people to be located in accessible places at certain times. Daily routine activities require people to leave the safety of their home and their family (relatives, friends) and bring them together with strangers of different background. The timing of work, school and leisure activities affects crime rates. Moving on dark and empty streets at night increases the suitability of a target [CF79].

The authors assembled diverse criminological analyses into a single framework, linking legal and illegal activities. In the area of descriptive analyses the authors assessed the work of Thomas Reppetto [Rep74], who provided evidence that an increased distance of households from city centers reduces risks of becoming a victim of a residential crime. Other studies report that city planning and community crime programs have the ability to decrease target suitability. They found strong empirical evidence that crime rates depend on the hour of the day and that offenders typically commit crime near their own homes. Regarding the macroeconomic perspective of crime trends, sociologists attempted to link crime rates to the economic conditions of the society with inconsistent results. Leroy Gould [Gou69] found that an increase of circulating cash and available cars caused an increase of bank robberies and car thefts. While the findings suggest that crime is opportunity-driven, the presented empirical evidence does not explain the relationship between routine legal activities and criminal acts. In the next step, the authors attempted to answer the question whether micro-level data is consistent with their routine activity approach [CF79].

Routine activities are defined as daily actions with the goal of satisfying basic needs including work, food provisioning, social interaction, leisure, entertainment and education. These activities occur at home, work and other places away from home. The authors stated that after World War II, people in the U.S. made a relocation of routine activities from home to work and other places. This relocation of routine activities results in an increase of convergences of the three elements of crime in space and time. This phenomenon contributes to the observed increase of crime rates. Cohen and Felson postulated that if the routine activity approach is valid, evidence must be found for three relationships. Routine activities near home and among family members are safer. Routine activities influence property availability towards visible and accessible places. The production of consumer goods rises parallel to crime rate trends [CF79].

In an empirical assessment, the authors first investigated the circumstances and locations of offenses. The data indicates that risk of victimization is dependent on the social distance between offender and victim. Another result is that the risk of becoming a victim alone is far higher than for groups. The risk of becoming a victim of a robbery is 105 times higher in streets than at home. It is assumed that Americans spend 16.26 hours per day at home, 1.38 on streets and 6.36 at other places. In regards to target suitability, one assumption is that the availability of expensive and movable goods such as cars, watches, electronic devices etc. have the highest risk of illegal removal. A comparison of the composition of stolen property reports by the FBI with national data on personal consumer expenditures for goods suggested that vehicles and electronic appliances are overrepresented in thefts. Another empirical assessment concerned the relationship between family activities and crime rates. According to the routine activity theory it is assumed that people in single households with a job outside home should have higher risk of being victimized. Similarly, adolescents and young adults, who spend more time in peer groups, have a higher risk of being victimized. In contrast, married couples are expected to have lower rates. The presented empirical evidence supports these assumptions. A conclusion is that the routine activity theory is consistent with the examined data [CF79].

Based on the empirical evidence, the authors presented the main hypothesis. It postulates that the increase in crime rates in the U.S. since 1960 is connected to changes in the routine activity structure of society. The changes of the society also facilitate an increase in suitable targets and a decrease in guardianship.

In order to verify their hypothesis, changes in human activities were investigated based on US Bureau of the Census Regulation (USBC) numbers from 1960 to 1970. The authors discovered increases of out of town travel, national and state park visits and vacations by 81 % from 1965 to 1972. There was also a significant increase of the female work force (31 %) and the number of single households increased by 34 %. Comparing hourly data of unattended households, an increase by 49 % at 8:00 a.m. was found. While the change of age structure of the U.S. population drastically slowed down at 1970, the changes of the routine activity patterns continued [CF79].

After establishing evidence for major social changes, the authors investigated related trends concerning property and consumer goods. Within the observed period (1960 to 1979), the sales of consumer goods increased, while size and weight decreased. Major declines in weight for radios, record players, televisions and other devices were noticed. These changes result in an increased target suitability for theft [CF79].

In the next step, Cohen and Felson applied their routine activity approach to model crime rate trends. A hypothesis was defined, stating that changes in crime rates in the U.S. are connected to the shift of activities away from home. For their study, the authors included the population aged 15-24 as first control variable and unemployment rates as second control variable. Five crime types were investigated: nonnegligent homicide, forcible rape, aggravated assault, robbery, burglary. The problem was modeled as difference equation. The first difference was modeled as dependent variable: the change in the official crime rate per 100 000 population between year $t-1$ and year t. In an alternative approach, the autoregressive form was formulated: the official crime rate in year t as a function of the exogenous predictors, adding to the official crime rate in year $t-1$. Both approaches were estimated by the ordinary least squares method and calculated for the years 1947 to 1974 [CF79].

In their findings, the authors reported that the time-series analysis confirmed a statistically significant relationship between the household activity ratio and changes in crime rates. The findings are significant for both the first difference form and the autoregressive form and for all types of crimes. Furthermore, the relationship is also positive no matter if age and unemployment control variables are included or not. A Durbin–Watson statistic also shows that autocorrelation is absent and the findings are robust and significant. The results suggest that routine activities have an impact on the occurrence of illegal activities [CF79].

Cohen and Felson concluded that if the convergences in time and space increase in number, then crime rates increase even without any changes in the structural conditions such as criminal propensity or number of offenders. Predatory crimes may be assumed as a side effect of freedom and prosperity much rather than social breakdown [CF79].

2.3.2 Applications and Implications

Based on the principles of the routine activity theory, other studies provided evidence that certain behavioral patterns increase the chance of becoming a victim of a crime. Risky behavior includes living in areas with higher crime rates, being on the street late at night, carrying valuable goods, being under the influence of alcohol and being alone without friends or family as guardians [Sie10].

Law enforcement organizations and city administrations have been deriving measures from the routine activity approach since its publication. One relevant result is the situational crime prevention. This strategy does not aim to reduce crime by punishment or resocializing of the offenders, but it aims to reduce the number of occasions where the three elements of crime converge. According to Cohen and Felson, a crime does not take place if one of the three elements is absent. In this sense, the goal of situational crime prevention is to reduce the opportunities for crime [Sie10].

Problem-oriented policing aims to identify root causes of crime in order to reduce crime. It was developed by Goldstein in 1979 and provides a framework for diagnosing causes of crime and developing a solution [ES87].

Michael Tonry [Ton95] identified three main measures of the situational crime prevention, which correspond to the three elements of the routine activity theory. The first measure aims to make crime more difficult, for example by gun control laws. The second measure aims to reduce the number of possible targets. The third measure aims to increase the risk for offenders of getting caught, for example by installing security cameras. In a later work, Clarke and Eck [CE05] expanded this approach and formulated 25 techniques of situational crime prevention. Ross Homel [Hom96] presented case studies of crime prevention techniques which implement the concepts of situational crime prevention. The measures in the area of city planning included improved lighting of public places, installation of locks and security systems, closed-circuit television (CCTV) surveillance cameras and access controls.

2.3.3 Criticism

The routine activity approach is a widely accepted theory in sociology and criminology. Crime prevention strategies based on this approach have been proven in studies and implemented successfully [Ton95]. Several studies have been conducted to evaluate the routine activity theory based on empirical data. However, due to the difficulty of obtaining personal data, the empirical validity is still debated [AS12].

The situational crime prevention is accused of merely relocating crime in a spatial, temporal and methodical dimension instead of preventing it. By focusing on the situational aspects of crime, the real underlying reasons of crime remain unregarded. A sustainable reduction of crime rates cannot be accomplished without acknowledging social circumstances such as poverty, poor education, drug abuse, unemployment, a lack of perspective, or discrimination. The routine activity approach, as a form of rational choice theory, exhibits the same inherent weakness. It assumes a perfectly rational thinking and deciding individual. The rational offender would suspend his criminal acts as a result of crime prevention measures. The rational choice theory does not account for emotional, social and psychological factors. Critics argue that the situational crime prevention requires a higher degree of surveillance, which violates basic human privacy rights. Another criticism is that crime victims are assumed with risky behavior and being unable to avoid the danger of becoming a victim [CE05].

The routine activity theory assumes an omnipresent criminal propensity of a certain part of the population. But human behavior is infuenced by emotions, moral weakness and provocations. Further influence exists due to peer pressure and the environment [Gar08]. Recent crime theories such as the sociological theory and the multifactor/integrated theory superseded classical theories [Sie10].

After an overview of related work, Chapter 4 describes the Street Crime Model, an agent-based simulation model derived from the main concepts of the routine activity theory.

CHAPTER 3

Context and Related Work

This chapter is structured according to the structure of this thesis. The first section presents literature concerning computer simulation as a scientific method. Selected papers, which address the methodology of this scientific method, are identified, followed by a selection of works where computer simulation is applied in scientific research. The following section specifically focuses the agent-based simulation approach. Available literature in the area of computer simulation as a research method for crime prevention and analysis is presented in the next section. The fourth section discusses available literature concerning the routine activity theory in social science. Related studies to this thesis are presented and the differences are highlighted. The scientific benefit of this thesis in the context of related work is explained. The final section discusses literature about the simulation-based research of hot spots of crime, which is the topic of the second research question of this thesis.

Computer simulation as a scientific method

The methodology of this work is based on the available litarature on computer simulation in science. Important contributions in this area are published by Axelrod [Axe97], Balci [Bal03], Bonabeau [Bon02], Gilbert [Gil96], [GT00], Happach and Tilebein [HT15], Kalua and Jones [KJ20] and Winsberg [Win19].

The possible applications of computer simulation in social science and economics are various as the following selection of research topics confirms. Siahaan et al. [Sia+17] researched „the development of students' science process skills". Huttar and Brintzen-hofeSzoc [sic] [HB20] conducted a literature review about computer simulation in social work education. In the area of economics and logistics, Straka et al. investigated the „Design of Large-Scale Logistics Systems Using Computer Simulation Hierarchic Structure" [Str+18]. Ciampaglia [Cia18] published the paper „Fighting fake news: a role for computational social science in the fight against digital misinformation".

In the field of agent-based modeling and simulation Macal and North [MN09], Squazzoni [Squ12] and Railsback and Grimm [RG12] published comprehensive summaries of the basics of the topic. Bonabeau [Bon02] researched „Methods and techniques for simulating human systems“. Gilbert and Terna [GT00] also identified and described scientific methods for agent-based models in social science. Several papers emphasize the integration of an agent-based modeling approach and GIS. Heppenstall et al. [HCS11] described „Agent-Based Models of Geographical Systems“. Further research of the authors resulted in the works „The Integration of Agent-Based Modelling and Geographical Information for Geospatial Simulation“ [CC12] and „The Use of Agent-Based Modelling for Studying the Social and Physical Environment of Cities“ [Cro12].

The following selected works represent the application of an agent-based modeling and simulation approach in science. In a social science study, Schuhmacher et al. [SBG14] simulated the development of risk behaviors during adolescence. Schenk et al. [SLR07] conducted simulation-based research about „consumer behavior in grocery shopping on a regional level“. A recent study by Dignum et al. [Dig+20] investigates the health, social and economic impacts of the Covid-19 pandemic by an agent-based simulation method. A related study by Silva et al. [Sil+20] evaluates the effects of social distancing policies during the Covid-19 pandemic.

Agent-based modeling and simulation software as a scientific tool

Repast is a widely used open-source agent-based modeling and simulation software. Several studies in different fields of science are identified where Repast is applied as tool for an agent-based research approach. Abar et al. [Aba+17] presented a systematic literature survey of the state-of-art in agent-based modeling and simulation tools. North et al. [Nor+13] described the development of Repast Symphony in their paper „Complex adaptive systems modeling with Repast Simphony“. Collier and North [CN13] worked on the Repast for High Performance Computing (Repast HPC) project to develop an ABMS system for high performance computing. Parry et al. [PEH06] worked on a method to run a single Repast model in parallel on a distributed network. Rui and Yong [RY17] conducted research on a rumor propagation model: „Modelling and Simulation for Rumor Propagation on Complex Networks with Repast Simulation Platform“. In the area of economics, Lou et al. [Lou+18] published the work „Behavior Simulation of Manufacturing Services in a Cloud Manufacturing Environment“. In the field of medicine, Bora et al. [Bor+17] published the paper „Modeling and simulation of the resistance of bacteria to antibiotics“.

This thesis describes the application of an agent-based simulation model for evaluation of spatio-temporal social phenomena. A simulation software package was selected to support both the agent-based and the geographic dimension of the research. Agent Analyst, based on the Repast software package, satisfies these demands. A documentation of Agent Analyst including a collection of case studies is provided by Johnston [Joh13]: „Agent-Based Modeling in ArcGIS“. An application of Agent Analyst in the energy sector is published by Imran et al. [ISM17] with the title „Agent-based simulation for biogas power plant potential in Schwarzwald-Baar-Kreis, Germany: a step towards better economy“. Munir et al. [Mun+20] published a study about the healthcare system in Pakistan simulating geospatial and socio-ecologic aspects in Agent Analyst: „Geospatial assessment of physical accessibility of healthcare and agent-based modeling for system efficacy“. In the area of biology, Bridge et al. [Bri+17] used an agent-based simulation to examine how the behavior of individuals scales up to give rise to population-level phenomena. Alghais and Pullar [ND17] developed an agent-based spatial model to investigate changes in land-use patterns based on forecast population estimates and planning policies in Kuwait. Tan et al. [THL15] used an agent-based approach to study building evacuation performance for increased safety: „Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency“.

Computer simulation as a research method for crime prevention and analysis

The application of computer simulation for crime prevention and analysis is a widely recognized research area. A systematic literature review on this topic has been conducted by Groff [GJT19]: „State of the Art in Agent-Based Modeling of Urban Crime: An Overview“. From 285,119 initially through electronic search identified studies, 45 studies retained after the comprehensive assessment. Selected papers from this list are „Generative Explanations of Crime: Using Simulation to Test Criminological Theory’“ by Birks, Townsley and Stewart [BTS12], „Social simulation and analysis of the dynamics of criminal hot spots“ by Bosse and Gerritsen [BG09b], „Generating crime data using agent-based simulation“ by Devia and Weber [DW13], „Modeling Civil Violence; an agent-based computational approach“ by Epstein [Eps02], „Nonlinear Dynamics of Crime and Violence in Urban Settings“ by Fonoberova et al. [Fon+10] and several works by Malleson et al. [MHS10], [MB12], [Mal+13].

Other studies in the field of agent-based modeling and simulation for crime prevention are published by Frank and Brantingham [BB95] [BB04] [Bra+05], Quijada [Qui+05], Malleson and Berkin [MB12], Bosse and Gerritsen [BG08] [BG09a], Groff [GM08], Squazzoni [Squ12], Macal [MN08] [MN10] [MSN04], Malleson [Mal+13] [MB12], North [Nor+13], Bonabeau [Bon02], Gilbert [Gil08] [Gil10], Ferber [FDP08], Lee [LY10], Crooks [Cro12] and Wise [WC14].

The routine activity theory in social science

The research questions are concerned about the work „Social change and crime rate trends: A routine activity approach“ by Cohen and Felson [CF79]. There are relevant obstacles involved in collecting empirical data of citizens and their daily routine activities in order to verify hypotheses in social science, for example privacy concerns and the lack of data of crime events at the individual level. As a result, only four attempts of providing empirical evidence of the routine activity theory are identified from the available literature. The first work is the study „Routine Activity and Victimization at Work“ by James P. Lynch [Lyn87]. Lynch showed that work-related activities have a greater impact on risk of victimization than socio-demographic attributes. Another work has been published by Mustaine and Tewksbury [MT99] with the title „A Routine Activity Theory Explanation for Women's Stalking Victimizations“. Mustaine and Tewksbury conducted a study among 861 college students from nine university faculties. Pratt et al. [PHR10] conducted a study among 922 adults in Florida titled „Routine Online Activity and Internet Fraud Targeting“. Finally, Maimon et al. [Mai+13] published the paper „Daily trends and origin of computer-focused crimes against a large university computer network: An application of the routine-activities and lifestyle perspective“. The empirical validity of the routine activity theory is still debated [AS12]. A possible solution is the application of computer simulation as a scientific method to produce empirical data [Win03]. One goal of this thesis is the generation of routine activity data in order to evaluate the main hypothesis of the routine activity theory.

The following studies are also concerned about evaluation of crime by application of the routine activity theory: „Agent based simulation of routine activity with social learning behavior“ by Amrutha [Amr14], „Spatial/Temporal Variations of Crime: A Routine Activity Theory Perspective“ by de Melo et al. [Mel+18], „Crime reduction through simulation: An agent-based model of burglary“ by Malleson et al. [MHS10] and „'Situating' Simulation to Model Human Spatio-Temporal Interactions: An Example Using Crime Events'“ by Groff [Gro07b]. Groff published another related paper in 2008 „Adding the Temporal and Spatial Aspects of Routine Activities: A Further Test of Routine Activity Theory“ [Gro08]. Some of the listed papers contribute to evaluating the routine activity theory by application of computer simulation, especially the works of Groff. The implemented model in the proposed thesis is more sophisticated in terms of agent decision making, agent movement and modeling of the environment. A more detailed model produces more accurate data and provides more trust with regards to the evaluation of the results. Another novelty of this approach is that crime hot spots and different policing strategies are evaluated relating to the routine activity theory. A literature survey of crime-related simulation studies for Maputo City returned no results. The evaluation of the routine activity theory combined with an African capital city as an object of study is a new approach and should lead to new scientific findings.

Simulation-based research of hot spots of crime

The second research question is concerned about evaluating policing strategies and relocation of crime regarding hot spots or clusters of crime. The displacement of crime is a relevant research area in criminology. For example Bowers and Johnson [BJ03] developed a method to measure the geographic displacement of crime. Several other works contribute to the topic, for example Braga et al. [BPH12] investigated the effects of focused police crime prevention interventions at crime hot spots, and whether law enforcement activities at specific locations result in crime displacement [Bra+19]. Weisburd et al. [Wei+17] conducted an agent-based simulation investigating the research question „Can hot spots policing reduce crime in urban areas?“. Zhang and Brown [ZB13] developed police patrol districting plans by using a parameterized redistricting procedure. Leigh et al. [LDJ19] also conducted a study about hot spot policing. None of the presented work is based on a routine activity model.

CHAPTER 4

The Street Crime Model

In this chapter, the simulation model is presented and discussed. The goal of the simulation model is to facilitate an answer to the proposed research questions in the introduction. An agent-based simulation model is built to evaluate the routine activity theory by Cohen and Felson [CF79] based on the geographic setting of the city of Maputo. A simulation model represents a simplified view of the reality. The investigated social phenomenon is street robbery in Maputo city. The subject of study was selected due to a research cooperation of TU Wien and Eduaro Mondlane University in Maputo. Cohen and Felson emphasized the spatio-temporal aspects of crime, so for a better understanding of the real world, a combined ABM/GIS approach is implemented.

In the first section, the main constructs and procedures of the simulation model are presented (Section 4.1). The second section explains how agent movement is modeled (Section 4.2). Agent rules are defined in the third section of this chapter (Section 4.3). In the final section, the parameters of the model are listed and discussed (Section 4.4).

4.1 Overview of the Simulation Model

The conceptual background of this simulation model is based on the criminological theory published by Cohen and Felson [CF79]. Their proposed routine activity theory (see Chapter 2.3) postulates that crime rates increase when the activities of people shift away from their home. The type of criminal acts under study were direct-contact predatory violations, or street robbery. Cohen and Felson identified daily routine activities of the population as a major factor for changes in crime rates. Routine activities such as work, shopping, sports or leisure facilitate the occurrence of convergences in space and time of the three elements of street robbery: the motivated offender, the suitable target and the absence of a capable guardian.

Based on this sociological model, an agent-based model is created. The presented street crime model is focused on the rules and behaviors of agents. Agents are autonomous decision-makers in dynamic interactions. Agents may interact with each other, or with the environment. The simulation model consists of two types of agents. Civilians are the first type of agents. Each civilian agent is unique in its personal set of routine activities. Civilians stay at home, or follow their routine activity route through the street network of Maputo. This behavior models the spatial aspect of the study. The temporal aspect is modeled by the passing of ticks during the simulation. One tick represents one minute of the real world and all agent decision routines and model procedures are based on those ticks. Routine activities are performed daily by every citizen. Macroeconomic concepts such as employment are modeled on a monthly basis and the simulation results are collected for at least one simulated year.

Civilians may act as one of three roles. A civilian can become an offender and commit acts of street robbery. Also, civilians may become a victim of a crime. Civilians incorporate the role of a guardian, when they witness an attempted crime and prevent it as bystanders. These are the three elements of street robbery defined by Cohen and Felson [CF79]. Police officers form the second type of agents in the street crime model. The purpose of police agents is to prevent crimes. Police agents do not follow their routine activities on a daily basis, but move around the street network.

Every agent owns a set of individual characteristics that represent its state. These characteristics are modeled as attributes, or fields, which can hold values of certain data types. In total, there 56 fields that characterize the citizen agents. Relevant attributes are name, activity spaces, wealth, income, criminal propensity, employment status, current location, or being at risk of a crime. These main attributes affect the decisions being made by the agents and have an impact on the macro-level of the simulation. Many other attributes exist for the purpose of collecting data for the simulation results. Police agents are less complex and they are characterized by 14 fields. Relevant attributes for police agents are name, current location, number of prevented crimes and the hot spot path as an array of street nodes. At the start of the simulation, agent attributes are initialized. This means that values are calculated from formulas and stochastic distributions and assigned to variables.

The environment in which the agents interact is based on the street network of the city of Maputo. The street network consists of street nodes, connected by streets. Routine activity spaces are modeled as special nodes within the environment. Agents always reside on one node and move from node to node. The streets that connect these nodes are not part of the simulated environment, but specify the neighbors of each node. Routine activity spaces are home, work and two locations for procurement and leisure activities. Routine activities affect the time spent in safe or unsafe places and movement patterns of citizens. As a result, changes in routine activities influence the number of space-time convergences among the main constructs. The next section provides more details about activity spaces and the movement of agents.

According to Cohen and Felson [CF79] a robbery occurs when three elements converge in

Human activity space	Condition for occurrence of crime
Home	Safe
Work (for employed citizens)	Safe
Street intersection	Not safe
Activity location 1	Not safe
Activity location 2	Not safe
Activity location 3 (for unemployed)	Not safe

Table 4.1: Overview of citizen agent activity spaces.

space and time. In the presented simulation model, a convergence takes place, when a motivated offender and a suitable target are situated at the same location at the same time. A motivated offender is a citizen with a certain composition of attributes, for example the attribute `criminal propensity` must be `true`. A suitable target is a citizen agent with certain characteristics, for example, the target must not have a lower wealth value than the offender. A convergence leads to a street crime, unless it is prevented by police officers or other citizens.

The characteristics of the environment are defined within society-level and environmental attributes. They also include input parameters for the simulation, which are a special case explained in Section 4.4. In order to increase the validity of the model, empiric data is used for population and employment attributes. The simulated street network is based on geographic data from OpenStreetMap. It is used for visual output of the simulation in ArcGIS, where the current state of agents can be represented visually on the city map during the simulation. In order to record simulation results for further analysis, tables for output data are created. Tables are saved in the CSV format and contain the final state of agents and of the environment.

The study of the second hypothesis concerning the emergence of crime hot spots requires certain modifications to the basic, or standard model. In the advanced model, the movement of police officers is changed to a systematic route. All other agent rules and input parameters, especially the RNG seed, remain unchanged.

4.2 Activity Spaces and Agent Movement

Daily routine activity spaces for each of the citizen agents as listed in Table 4.1 consist of different home, work and activity locations. At the start of every simulated day, each citizen agent starts its everyday routine activities at its home node. The home activity space is safe, so no robberies can occur. The duration a citizen agent spends at home depends on the simulation parameter `percentage of day safe`, which is systematically varied to investigate its impact on the number of convergences and committed robberies.

When the calculated time at a specific activity space has passed, the citizen agent starts moving along its activity space route until the next activity space is reached or the day is over. At the end of each day, all citizens are reset to their home activity space. By moving agents along their activity space route, traveling on the streets is simulated. Different types of travel like on foot, riding a bicycle or driving a car are modeled by generating random numbers from a uniform distribution. In effect, the travel speed is calculated randomly for each move. While citizen agents travel through the street network, they are at risk of being robbed.

After leaving their home node and traveling along the activity route, employed citizen agents eventually reach their work node. Unemployed citizens visit their activity place three instead. The work space is the main activity location away from the household and represents the main occupation of the citizen including job, school and university. The work activity space is safe, so no robberies can occur. The duration of the work activity varies for each citizen and it is modeled by generating random numbers from a normal distribution with constant parameters.

When the time at the work activity space is over, the agents continue their path along the activity route until reaching activity place one and finally activity place two. These places represent many essential daily activities including the acquisition of nutrition, shopping goods for daily needs, going to a physician or a hairdresser, maintaining social contacts, exercising sports and visiting locations for entertainment and leisure. All these daily routine activities are summarized as activity places one to three. Activity places are not safe, so citizens are at risk of being robbed during their stay.

The movement of citizens is directed and based on pre-computed paths. The Agent Analyst package does not support the routing features of ArcGIS, so the street intersections are implemented as node layer. Activity spaces are computed for every citizen as list of nodes prior to the simulation as described in Chapter 5.3. The list of nodes defines the shortest path to be traversed to visit all four activity spaces.

The movement of police agents is implemented in two types. In the standard model, agents move randomly across the Maputo city street network. In the advanced model, one half of the police officers move along a predefined route, connecting the identified crime hot spots from the standard model, while the other half still moves randomly. While every citizen agent has its own unique activity space list, police agents share a common list of nodes defining the hot spot route. Every police officer on hot spot duty is initially placed along the hot spot route at a random position. The random movement of police officers is modeled in two steps. In the first step, the current node of the police officer is used to obtain a set of adjacent nodes. In the second step, one target node is selected randomly and the target node is the new current node of the police agent. The neighbors, or adjacent nodes, are not determined at run time, but pre-processed and saved in a file. This approach is called a *random walk*.

4.3 Agent Rule Definitions

The behavioral background of agents in the context of social simulation has been presented in Chapter 2.2.5. In the Street Crime Model, the behavior of citizen and police agents is based on simple decision rules. Nodes are also considered agents, but of the *VectorAgent* type. There are also rules that concern the macro-level of the simulation, for example rules concerning income and employment.

All citizens move along their personal pre-computed routine activity paths. Daily routine activity spaces are home, work, activity place one and activity place two with movement along streets in between the activity places. Citizens are not at risk of being robbed at home or at work. Offenses may occur during traveling along the activity path or staying at activity locations one to three. The movement speed of citizens is a uniform distribution of one to six nodes per tick.

A robbery, according to routine activity theory, happens when the following requirements are met. There must be at least two citizens at the node in question and at least one of them must have the attribute `CriminalPropensity` with the value `true`, who becomes the offender. Both the offender and the victim are not located at a safe place. See Table 4.1 for a list of safe and unsafe locations. A stochastic function adds randomness to the calculation and is used to calibrate the resulting crime rates to the numbers found in empiric data. The victim must be a suitable target with a higher current wealth value than the offender. If there is more than one possible victim, the citizen with the highest wealth from all present citizens is chosen. The offender must be motivated to commit an offense. This is calculated from the `TIME_NEXTCRIME` attribute, which is individual for every offender. It means that after a successful robbery, the offender has to wait an average of two hours until the next offense. This is achieved by a distribution and ensures that offenders cannot commit an unrealistic amount of robberies each day. At this point, a convergence is registered based on the location of the victim. Another requirement states that no guardians may be present. The first check for guardians determines if there are police officers present. If that is the case, the robbery is prevented by a cop and the offender is set inactive for one to seven days. The second check for guardians concerns citizen bystanders. A formula generates a guardianship factor by the number of witnesses and a random factor as a uniform distribution from -3 to 3. If the result is lower than one, then not enough witnesses are present and the crime is committed. If the result is one, then a random decision follows with a 50 % chance for a robbery. If the result is higher than one, then too many witnesses are present and the crime is prevented by citizens. When it is decided that a robbery happens, the amount of money robbed is determined. The amount is a random number between one and either five or the current wealth of the victim, whatever is smaller. This model allows a maximum of one robbery per tick per node.

In the Street Crime Model, every citizen agent has a wealth attribute, which holds an integer number. The wealth value is relevant for the decision-making of a robbery as explained in the previous section. Income is a fixed amount of money units every

Parameter	Data Type	Default Value	Description
CITIZENS	int	1000	Number of citizen agents
POLICE	int	200	Number of police officers
NUM_PLACES	int	7001	Number of places
MODELTYPE	int	0	Simulation type
TIME_MODE	java.lang.String	Y	Simulated day, month or year
TIME_MULTIPLICATOR	int	3	Multiplicator for TIME_MODE
TIME_SAFE	double	74.75	Percentage of day spent in safe places
TIME_WORK	int	420	Minutes of a day spent at work
UNEMP_RATE	double	24.5	Unemployment rate in percent
TIME_NEXTCRIME	int	120	Duration until next crime
PROPENSITY	int	10	Rate of criminal propensity
INCOME	double	19.5	Income for citizens
WEALTH	int	475	Initially distributed wealth for citizens
HOTSPOTTIMER	int	10	Duration of stay on hot spot
RngSeed	int	100	Random number seed

Table 4.2: Parameters of the Street Crime Model.

employed citizen receives every 15 days. The simulation scheduler is responsible for these recurrent actions. For income distribution, the income value is added to the wealth attribute of the citizen. Every 30 days each unemployed citizen agent has a 5 % chance to become employed and the same number of employed citizens become unemployed.

4.4 Parameters of the Model

In this section, all relevant model parameters are presented. Table 4.2 shows a summary with data type, value and description of all default model parameters. If possible, empiric data was acquired to improve the validity of the model. When no empiric data was found, assumptions were made and the considerations are explained for each parameter.

CITIZENS specifies the number of citizen agents for the simulation. Citizens may act as offender or as victim, but are not part of the police force. The simulation target is the city of Maputo with a population of 1.104 Million in 2020 [Uni20]. Due to modeling and performance restrictions, the number of simulated citizen agent is 1000. As a result one simulated citizen agent represents 1104 inhabitants in the target system.

POLICE specifies the number of police officer agents for the model. Police officers never commit crimes and cannot be target of a crime. They do not pursuit daily routine activities like citizens. The number of police agents is set to 200 due to the random movement of police agents. Lower numbers would result in almost no prevented offenses.

`NUM_PLACES` specifies the number of place nodes. The simulated environment consists of 7001 nodes, which act as street intersections. The street network of Maputo is extracted from OpenStreetMap as a shapefile.

`MODELTYPE` defines the performed simulation type. The model type 0 represents the standard model and 1 represents the advanced hot spot model.

`TIME_MODE` controls the simulated time frame. The letter `D` represents day-length simulation time, `M` stands for month and `Y` for year.

`TIME_MULTIPLICATOR` is a factor concerning the time dimension of the simulation. This number multiplies the set `TIME_MODE` in order to achieve various simulated time frames from one day to N years.

`TIME_SAFE` represents the percentage of the day spent in safe places. The `TIME_SAFE` value is the sum of time spent at activity place home with a variable duration and the activity place work with a fixed duration. Unemployed citizens do not visit a work place, so their average `TIME_SAFE` is lower than of employed citizens. The time safe value of 68 % found in empiric data[1] is approximated by an input value of 74.75. The discrepancy is explained by unemployment mechanics.

`TIME_WORK` specifies the duration an employed citizen spends at work (or educational place) each day. Time spent at work is modeled as a normal distribution with mean $\mu = 420$ and standard deviation of $\sigma = 60$. An average of seven working hours per day is assumed.

`UNEMP_RATE` defines the unemployment rate in percent. Unemployed citizens do not visit a work place. Every month, 5 % of all unemployed citizens become employed and the same number of employed citizens become unemployed to keep the unemployment rate constant. The source for the unemployment rate is *The World Factbook* by the Central Intelligence Agency (2020) [Cen20].

`TIME_NEXTCRIME` defines the duration in ticks an offender cannot commit another crime after a successful robbery. This value is set at a default value of a normal distribution with mean $\mu = 120$ (two hours) and standard deviation of $\sigma = 12$.

`PROPENSITY` holds the rate of criminal propensity in percent. The parameter `PROPENSITY` is modeled as a uniform distribution among all citizens. The source for the criminal propensity rate is *Participation in criminal careers* by Visher and Roth [VR86].

`INCOME` is a fixed amount of money units every employed citizen receives every 15 days. The amount of income is derived by the GDP per capita 474.50 USD (2018), which corresponds to 19.50 every 15 days. The income is approximated by a normal distribution

[1] Cohen and Felson [CF79] determine the average time spent away from home to be 7.74 hours per day, which corresponds to 68 % of a day spent in safe places.

with mean $\mu = 19.5$ and standard deviation of $\sigma = 19.5/4 = 4.875$. The source for the `INCOME` parameter is *IMF Data* by the International Monetary Fund [Int18].

`WEALTH` corresponds to the amount of money a citizen agent owns. The initial wealth is calculated using a normal distribution with mean $\mu = 475$ and standard deviation of $\sigma = 950$. If the normal distribution results in a negative value, then the initial wealth is set to one. The source for the `WEALTH` parameter is *IMF Data* by the International Monetary Fund [Int18].

`HOTSPOTTIMER` specifies the duration in ticks (minutes) that police officers stay at a hot spot location before proceeding to the next hot spot along their route. This input parameter is assumed with a value of ten minutes.

`RngSeed` defines the seed input value for the random number generator. The seed value ensures that repeatable random numbers are generated during the simulation, so it is possible to replicate model results over multiple runs even though random number generators are used. This is a crucial requirement to using a specific simulation package as a scientific tool.

The derivation and the rationale of the model input parameters conclude the description of the Street Crime Model. The following chapter documents the used tools, preparations, activities and implementation steps involved in building and running the Street Crime Model.

CHAPTER 5

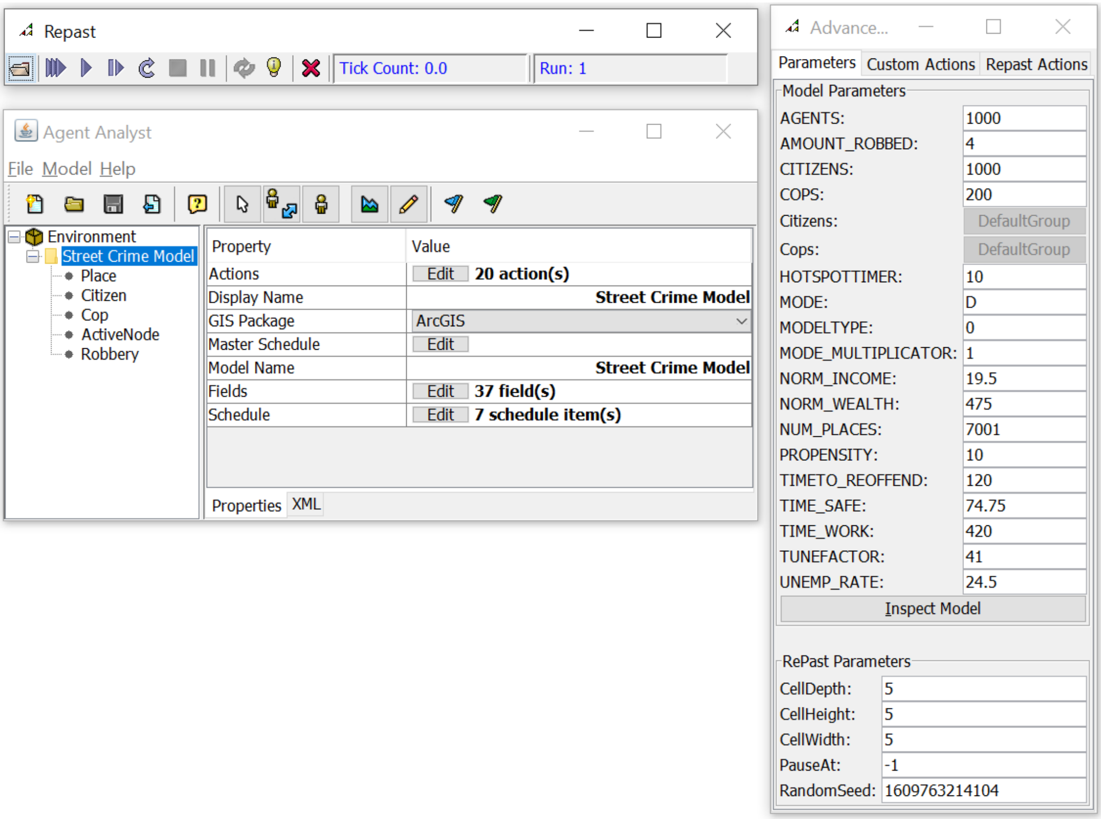
Agent-Based Simulation of the Street Crime Model

In this chapter, the implementation of the previously presented Street Crime Model and the development of the simulation program are described. In the first section, the tool landscape of the used GIS and ABMS software is documented (Section 5.1). In order to work with geographical data from OpenStreetMap in ArcGIS, the GIS data has to be prepared and processed (Section 5.2). The activity spaces and routes for 1000 citizen agents and the hot spot route for police agents are prepared in the next step (Section 5.3). The simulation model is implemented in a computer program using the programming language NQPy (Section 5.4). The certification that a simulation model is valid for a specific purpose is an essential task within the simulation model development process. The set of the applied certification measures consists of the verification (Section 5.5) and validation (Section 5.6) processes. The sensitivity analysis shows the reaction of the simulation output to the variation of a certain input parameter (Section 5.7). This method is used for testing the robustness of the model, calibrating parameters and it contributes to building trust in the results.

5.1 Integration of GIS and ABMS Software

ArcGIS ArcMap 10.1 for Desktop (see Chapter 2.1.6) is used in conjuction with Agent Analyst[1], a third party extension for ArcGIS. Agent Analyst integrates ABMS and GIS to enable agent-based simulation with both space and time dimensions. Agent Analyst v. 1.0b is based on the Repast Py toolkit (see Chapter 2.2.8). ArcGIS integrates Agent Analyst as a toolbox as a part of the ArcGIS geoprocessing environment [Joh13]. First,

[1] Agent Analyst: Agent-Based Modeling in ArcGIS. Esri, Inc: https://resources.arcgis.com/en/help/agent-analyst/ (visited on 01/05/2021)



Property	Value
Actions	Edit 20 action(s)
Display Name	Street Crime Model
GIS Package	ArcGIS
Master Schedule	Edit
Model Name	Street Crime Model
Fields	Edit 37 field(s)
Schedule	Edit 7 schedule item(s)

Model Parameters

AGENTS:	1000
AMOUNT_ROBBED:	4
CITIZENS:	1000
COPS:	200
Citizens:	DefaultGroup
Cops:	DefaultGroup
HOTSPOTTIMER:	10
MODE:	D
MODELTYPE:	0
MODE_MULTIPLICATOR:	1
NORM_INCOME:	19.5
NORM_WEALTH:	475
NUM_PLACES:	7001
PROPENSITY:	10
TIMETO_REOFFEND:	120
TIME_SAFE:	74.75
TIME_WORK:	420
TUNEFACTOR:	41
UNEMP_RATE:	24.5

RePast Parameters

CellDepth:	5
CellHeight:	5
CellWidth:	5
PauseAt:	-1
RandomSeed:	1609763214104

Figure 5.1: The graphical user interface of Agent Analyst.

the shapefile of Maputo is loaded in ArcMap. Then, on activation of the toolbox, the graphical user interface of Agent Analyst initializes. Now, the model in Agent Analyst is able to access the shapefile in ArcGIS. As a result, agents do have spatial features and it is possible to visualize agent movement and actions, for example the locations of crimes.

A screeenshot of the Agent Analyst middleware is provided in Figure 5.1. The graphical user interface of Agent Analyst facilitates the creation of agents and provides built-in functions for defining properties and actions. The scheduler enables users to call functions at certain intervals in order to model recurring events. The function responsible for visual output is also called by the scheduler every 1440 time steps to display daily results. Agents are able to query spatial data from ArcGIS and base their decisions on their surrounding world.

The input parameters can be viewed and modified in a separate window. Agent Analyst provides the control bar from Repast for starting, pausing and stopping the simulation. Support for batch runs is provided by the „Multiple Runs Start“ feature. Users are able to define various input parameters and sequences of parameters in configuration files, which serve as input for batch runs. This method is used for sensitivity analysis and for running the twelve scenarios for hypothesis testing.

5.2 Preparation of GIS Data

A computer simulation gains validity when it operates on real world data. One goal of this work is to simulate interaction of citizen agents in a realistic urban environment. This emphasis on the geographic dimension makes it necessary to import real mapping data from the city of Maputo, Mozambique. This section describes the process of obtaining spatial data and importing this data to ArcMap.

The process starts with pointing the web browser to `openstreetmap.org` and searching for the string „Maputo“. The zoom in/out tools can be used to select the exact sector of the map to be exported. There are several ways to export OpenStreetMap data. The default export function results in downloading an OSM file. An OSM file contains spatial data in the OpenStreetMap (OSM) format. The OSM file type is XML-based and stores three types of spatial data: points, connections and object properties. Large areas with more than 50 000 nodes can be downloaded via overpass API, or via third-party websites. It is not possible to directly import OSM files to ArcGIS. OSM files need to be converted to the shapefile (see Chapter 2.1.4) format. The conversion can be achieved by several tools: OSM2SHP[2], QGIS [QGI17] or MyGeodata Converter[3] as an online tool. There are also services generating custom shapefile downloads such as HOT Exports[4], BBBike.org[5] or OSMaxx[6].

During the following steps, the downloaded data is imported in ArcGIS and processed in order to generate a street network shapefile and an adjacent nodes list. In the first step, the downloaded shapefile with street data is opened in ArcMap. The ArcMap Editor can be used to clean the street network from parts that are not connected. The toolbox „CreateNodeList“ provided by the Agent Analyst package [Joh13] is used to convert the shapefile with street data into a shapefile of nodes. This is achieved by an ArcGIS geoprocessing model. A built-in graphical model builder allows the subsequent usage of tools, using the output of one tool as the input of the following tool. The „CreateNodeList“ model requires two parameters. The first required parameter is the streets shapefile as the input feature class and the second parameter is the street intersections shapefile as the final output.

The following process uses the result of the previously generated street nodes shapefile to define the set of adjecent nodes for each node. This is necessary for realizing the random

[2] OSM2SHP: Small application to convert from OpenStreetMap format (OSM) to shapefile format. `https://code.google.com/archive/p/osm2shp/downloads` (visited on 01/05/2021)

[3] Convert OSM to SHP Online. `https://mygeodata.cloud/converter/osm-to-shp` (visited on 01/05/2021)

[4] HOT Exports: OpenStreetMap data export tool. `https://export.hotosm.org/de/v3/` (visited on 01/05/2021)

[5] BBBike.org: OpenStreetMap data export tool. `https://extract.bbbike.org/` (visited on 01/05/2021)

[6] OSMaxx: OpenStreetMap data export tool. `https://osmaxx.hsr.ch` (visited on 01/05/2021)

Place Id	Point X	Point Y
2773	"3629044,11987000000"	"-2991537,66371000000"
5829	"3630948,32748000000"	"-2984974,86485000000"
5702	"3630986,67663000000"	"-2985355,14412000000"
4632	"3630005,33043000000"	"-2988441,48095000000"

Table 5.1: Activity nodes for the citizen agent with the name „a1“.

walk movement of police officers in the Street Crime Model. In order to create a set of adjacent nodes, the street intersections that are connected to each other have to be identified. A Python script tool is started from ArcMap, which creates a CSV text file containing all nodes with a list of adjacent nodes. The Python script accepts three input parameters: the node shapefile created in the previous step, the street network and the definition of the unique field, which corresponds to the unique place id of each node. The script traverses each node in the shapefile, identifies its neighbors and writes the information to the CSV file.

5.3 Creation of Activity Spaces and Routes

In the following steps, the personal four activity spaces for each citizen agent are created by a Python script. The capability of a GIS software is used to solve a route between those four spaces. The resulting list of nodes defines the path, on which a citizen agent moves in order to perform daily routine activities.

First, the dBASE table file as part of the nodes shapefile is opened in LibreOffice Calc and saved in CSV format. A Python script has been written to generate activity spaces for each of the citizen agents. The contents of an example file are displayed in Table 5.1. The script first reads the nodes CSV file and stores `point x` and `point y` coordinates for each node in two arrays. Then, for each of the 1000 agents four nodes representing home, work, activity one and activity two are randomly generated. Node ids range from 1 to 7001. The assigned node id is used to get the correct X, Y coordinates from the arrays and finally the data is written to the CSV file for each agent.

The following process describes the creation of lists of nodes connecting the previously generated activity spaces. This is achieved by programming two Python scripts and an ArcGIS geoprocessing model in order to automate the task of creating 1000 lists of nodes. First, a GeoDatabase is created an ArcGIS. Then, a network dataset is created from the GeoDatabase and the previously created street network shapefile and the node shapefile are added as feature datasets. The network dataset features such as route solving are provided by the Network Analyst extension of ArcGIS. The first Python script converts the 1000 CSV files containing the four activity spaces into shapefiles. The second Python script subsequently takes these shapefiles as input, calls the „Solver“ tool to generate the route between the four activity spaces and saves the results to ArcGIS Layer files. Finally,

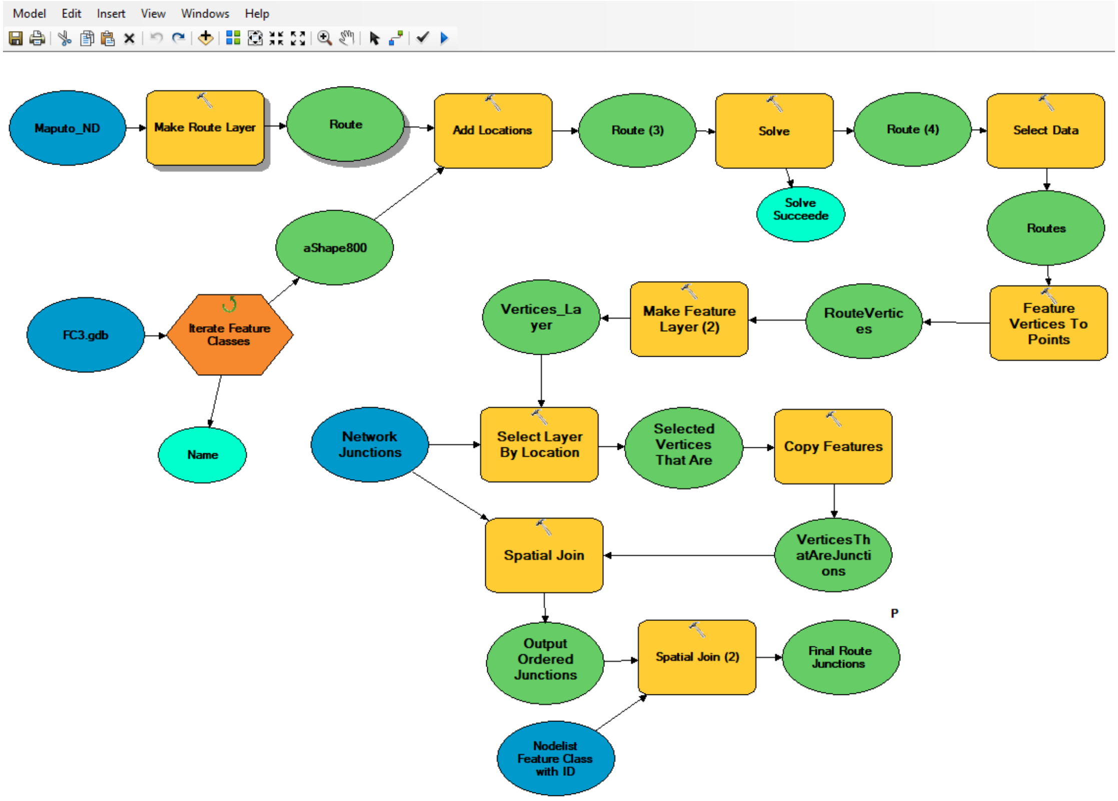


Figure 5.2: The geoprocessing model that generates the routine activity path for each agent. Adapted from [Joh13].

the ArcGIS geoprocessing model iterates through the 1000 layer files and converts them to shapefiles. The applied geoprocessing model is displayed in Figure 5.2.

One thousand activity path files have been created from node files to generate a route for each citizens. Along this route, each citizen starts at the home activity place and in turn visits the work node and the two activity places that represent for example shopping, social contacts and leisure.

5.4 Implementation of the Model

This section documents the implementation of the Street Crime Model in NQPy. The program code is directly entered in Agent Analyst, which only provides basic text editor features and lacks the feature set of a modern IDE. The full program code can be imported and exported to Agent Analyst in an XML-based file format. The source code for the Street Crime Model is available on Github[7].

[7] The Street Crime Model source code. https://github.com/philippschnatter/Street-Crime-Model.git (visited on 03/19/2021)

At the start of every simulation run, the model must be initialized. On start, the method `initAgents()` is called, which orchestrates all setup processes. First, it sets the random number generator seed and initializes distributions. Subsequently other methods are called, which control one part of the initialization process. The method `initModel()` initializes model level variables. The method `setupPlaces()` reads the street nodes from the shapefile and initializes a hashmap with place agents. Then, the file containing adjacent nodes is read and the neighbors are added to each place in the hashmap.

The method `initCitizens()` initializes citizen agents and sets initial values for agent attributes. The four activity spaces and the list of path nodes are set. Further, the durations to stay at home and at work are calculated and assigned. The employment status for the agent is defined. The criminal propensity attribute is set `true` for randomly chosen 10 % of all citizen agents. Finally, income and wealth values are derived from a normal distribution for each citizen agent. The method `initActivitySpaces()` reads the lists of the previously generated activity nodes and path nodes for each citizen agent.

The method `initCops()` initializes the police agents and sets initial values for agent attributes. In the advanced model, 100 of 200 police agents are assigned a hot spot route. The method `initCopRoutes()` reads the list of hot spot path nodes.

After the initialization process, the simulation scheduler dictates the sequence of actions. The `incrementModel()` method is scheduled to run at every tick. The method iterates through nodes with citizens and logs the presence of agents in the shapefile. Then it calls the `determineRobs()` function, which evaluates which agents are located on the occupied node and then determines whether a crime takes place. The actions of this method are documented in the model description in Chapter 4.3. The `incrementModel()` method further increases the model counter and recognizes the defined end of the simulation. At the last model step, the resulting shapefile is updated and output files for citizen agents, police agents, places and daily report containing the final state of variables are written to the file system. The `step()` methods of citizen and police agents are also executed at every tick. These methods realize agent movement. Citizen agents stay at an activity space, or move along their path to the next activity space. Depending on the current location, the `atRisk` variable is set.

There are several other methods invoked by the scheduler at certain intervals. The `updateDisplay()` method is executed at an interval of 1440 ticks, which corresponds to one day. This method refreshes the map in ArcGIS with current agent data for visualization. A the end of each model day, `resetCitizens()` places all citizen agents back at home, so they are set at the first node of their path, not moving and not at risk. The scheduler invokes `distributeIncome()` every 21 600 ticks (15 days), which adds the income value to the wealth value for all employed citizens. The `changeEmployment()` method is executed at an interval of 43 200 ticks (30 days) and creates a 5 % fluctuation at the job market.

5.5 Verification

Verification is the process of ensuring a high degree of software correctness. For the Street Crime Model, various techniques were applied to guarantee that the source code is working as intended. The scientific background on verification is presented in Chapter 2.2.7. Verification is an iterative process and it was performed during implementation of the model. It included running tests, identifying program errors and implementing corrections to the code.

The standard model was specified according to the routine activity theory as the Street Crime Model (see Chapter 4). It contains the definition of the interacting agents and their behavior as rules, actions and attributes. The environment where the agents interact was defined as the city of Maputo. Finally, agent-level and society-level parameters were determined according to empiric data, or by reasoned assumptions. The computer program was systematically tested against this specification.

Structured code walkthroughs were performed. Since Agent Analyst neither provides a debugger nor unit tests, the focus of verification efforts was set on model logging. Every method was tested by tracing its variables in log outputs. The actions of random citizen agents were logged step-by-step in order to verify the correct behavior. The same technique was applied to police agents. The circumstances of robberies were logged in order to verify the proper operation of the code. The log data in the context of verification was written to console or to a text file. The log also contained metadata about the model duration, whether the standard or the advanced model was run, and the `TIME_SAFE` input parameter. It can be concluded that Agent Analyst as middleware between ABMS and GIS software does not support important concepts of modern software engineering such as unit tests. As a result, a high effort was required for the verification of the program code due to the manual testing nature of code walkthroughs and output logging.

5.6 Validation

The validation process (see Chapter 2.2.7) aims to ensure that the simulation is a well-suited model of the target. The model is expected to reproduce the behavior of the target according to the research questions. The validity of the model is asserted by comparing simulation results with observed data from the target.

According to the methodological approach presented in Chapter 1.4 of the introduction, the task of validation corresponds to the steps five and seven: internal and external validation. Internal validation of the implemented model describes the process of comparing the simulation behavior to the theoretical model. External validation describes the process of comparing the simulation results with empiric data from the target system.

Two central variables are chosen for the validation process. An exogenous and an endogenous variable are selected from empirical data . The `TIME_SAFE` parameter is an exogenous variable and it represents the percentage of the day spent in safe places by a citizen agent. The TIME_SAFE value of 68 %, or 7.74 hours per day, was determined by Cohen and Felson [CF79] in their studies of the American society in 1979. The other variable is endogenous and denotes the observed crime rate per day. The crime statistics by the Instituto Nacional de Estatística – Moçambique reports 5828 street robberies for Maputo City in the year 2015 [Ins15]. The corresponding number of inhabitants of Maputo City is 1.187 Million (2015) [Uni20]. Normalized to 1 Million citizens, the crime rate is 13.45 street robberies per day.

First, the internal validity of the simulation is asserted. The key aspect of the conceptual model, as defined by the routine activity theory, postulates that crime rates increase when the activities of people shift away from their home. This means that when the input parameter TIME_SAFE is changed to lower values, the resulting crime rates should increase. This behavior can be reproduced by the simulation as expected. The resulting crime rates for systematically changed TIME_SAFE input are displayed in Chapter 6 „Results“ in Table 6.8.

Second, the external validity of the simulation is asserted. This is achieved by asserting that the simulation reproduces empirical data. The endogenous crime rate is compared. The results are produced by a simulation run with a default setting of input parameters. The simulation results are documented in Chapter 6.1. The average daily crime rate of the simulation (12.06) is comparable to the crime rate found in empiric data with a normalized value of 13.45.

The performed verification and validation procedures are aimed to build trust in the simulation model. Ultimately, the model must be valid in order to answer the two proposed research questions.

5.7 Sensitivity Analysis

Sensitivity analysis describes the process of changing model parameters and observing how the simulation results change. One parameter at a time is varied, while the values of all other parameters remain constant. By defining a fixed random number generator seed, Agent Analyst ensures that also results of random number generators are repeatable. Sensitivity analysis plays an important role in simulation-based research. It is a common practice for discovering the most important parameters for system behavior on which the simulation effort should focus on. Sensitivity analysis can offer an improved insight on how the variable affects the system behavior. Another application of sensitivity analysis is to certify the robustness of the model. A detailed summary of sensitivity analysis is presented in Chapter 2.2.7.

Sensitivity analysis was performed repeatedly as integral part of model verification and testing of the Street Crime Model. For documentation purposes, a sensitivity analysis was performed with the final model and with the final input parameter values as defined in Chapter 4.4. The four exogenous variables for which the sensitivity analysis was performed, are unemployment rate, minimum time between offenses, criminal propensity

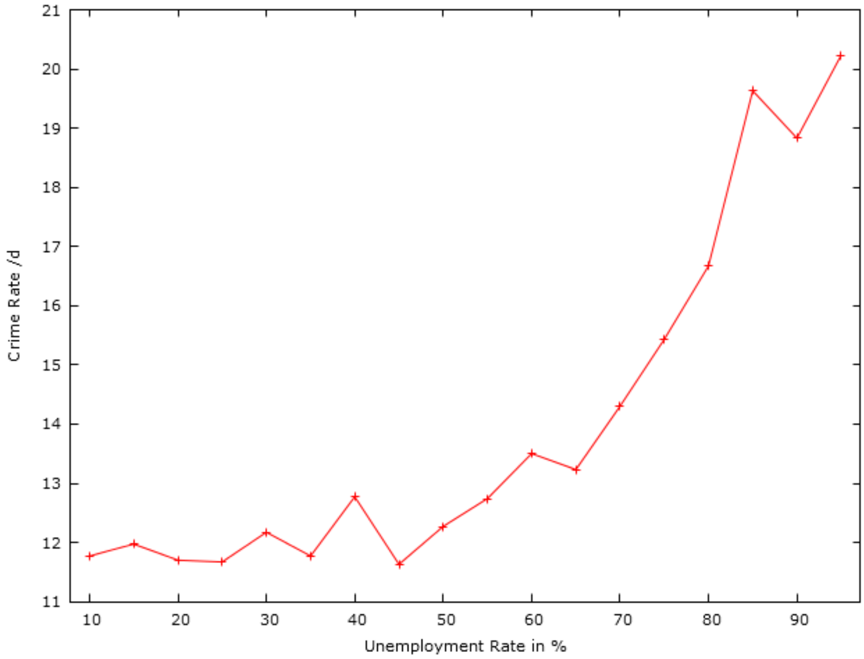


Figure 5.3: Sensitivity analysis of the exogenous variable unemployment rate.

and random number seed. The selected endogenous variable, on which the comparisons are based, is the observed crime rate per day. The crime rate per day is defined as the total number of offenses that occurred in one month (30 days) divided by 30.

The sensitivity analysis of the exogenous variable unemployment rate shows that the model is robust to changes in this parameter. The results are displayed in Figure 5.3. The data clearly indicates that crime rates increase when the rate of unemployment increases. This behavior is expected, because unemployed citizens spend more time in unsafe places, increasing the opportunities for street crimes.

The sensitivity analysis of the exogenous variable criminal propensity indicates a strong influence on crime rates. Figure 5.4 suggests a linear relationship with a strong positive correlation. This behavior is supported by the model. A motivated offender is defined as one of the three elements of crime according to the routine activity theory [CF79]. With an increasing number of offenders, a proportional increase of the number of offenses is observed.

The analysis of the exogenous variable minimum time between offenses indicates that the model is robust to changes in this parameter. The resulting crime rates remain stable for minimum time between offenses between one minute to one day. A further increase of this parameter results in a significant decrease of crime rates. A higher value of this parameter leads to higher intervals between possible offenses and ultimately to a lower number of offenses.

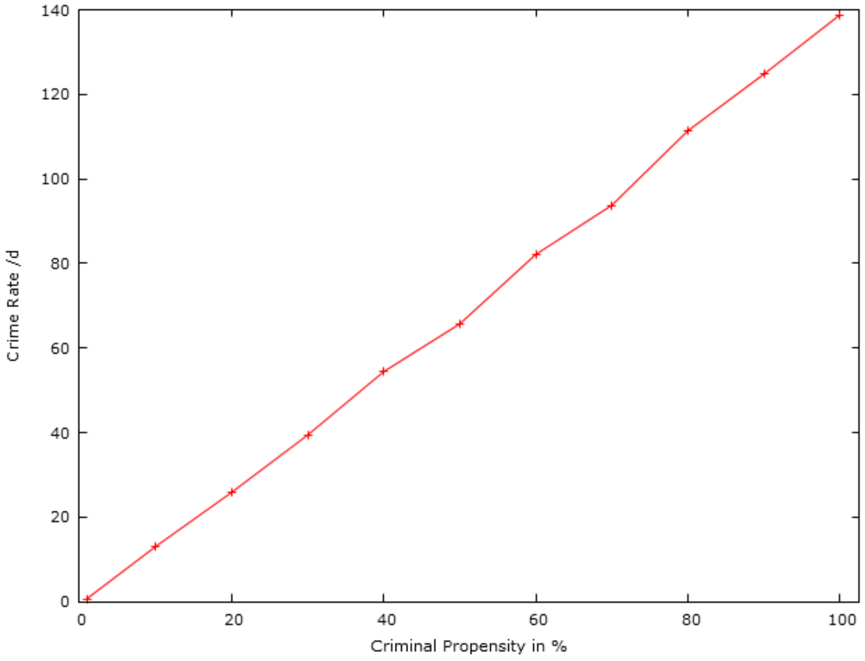


Figure 5.4: Sensitivity analysis of the exogenous variable criminal propensity.

The simulation of twelve scenarios for varied `TIME_SAFE` input parameters, which is performed in the upcoming Chapter 6, is also an application of parameter sweeps. The results are listed in Table 6.8. Further experiments showed that the model is robust to changes in the random number seed. This is a requirement for the validity of the results in the following chapter where the simulation results and research findings are presented.

CHAPTER 6

Results

In this chapter the results of the simulation are presented. The results of a simulation run consist of visual output from ArcGIS and five files in CSV format, which collect data from the variables of model agents. Multiple simulation runs with varied input parameters have been conducted to obtain the results in order to answer the two research questions. One simulation run with a time dimension of three years with default settings provides data for general model results and descriptive analysis (Section 6.1). The simulation results of the advanced model are presented in the following Section 6.2. In order to investigate the hypothesis of the routine activity theory, twelve simulation runs of one year each have been conducted. The study of the second hypothesis concerning the emergence of crime hot spots is facilitated by a simulation run of the advanced model with a duration of three years and otherwise default settings. In the final section, the two proposed hypotheses are evaluated using statistical methods (Section 6.3).

The primary output of the conducted simulation is the visual output in ArcGIS as shown in Figure 6.1. The street network of Maputo is displayed in the center of the application. Street intersections, or nodes, are highlighted, when offenses occur. The output is refreshed at every 100 ticks during the simulation and Figure 6.1 displays the final state at the end of the simulation with a time dimension of three years. It is clear that offenses are not spread evenly across all street nodes, but rather follow a pattern. The nodes follow a classification of ten groups in natural breaks (*Jenks*[1]). There are a few nodes with a high number of registered offenses, forming clusters, while whole areas of the city remain blank. One explanation of this emergent phenomenon are the daily routines of the 1000 citizens. They traverse along their daily activity spaces on the shortest path, forming streets with a high frequency of visits. Several „highways“ that run from north to south can be identified.

[1] The Jenks natural breaks classification method is a data clustering method designed to determine an optimal arrangement of values into different classes [Jen67].

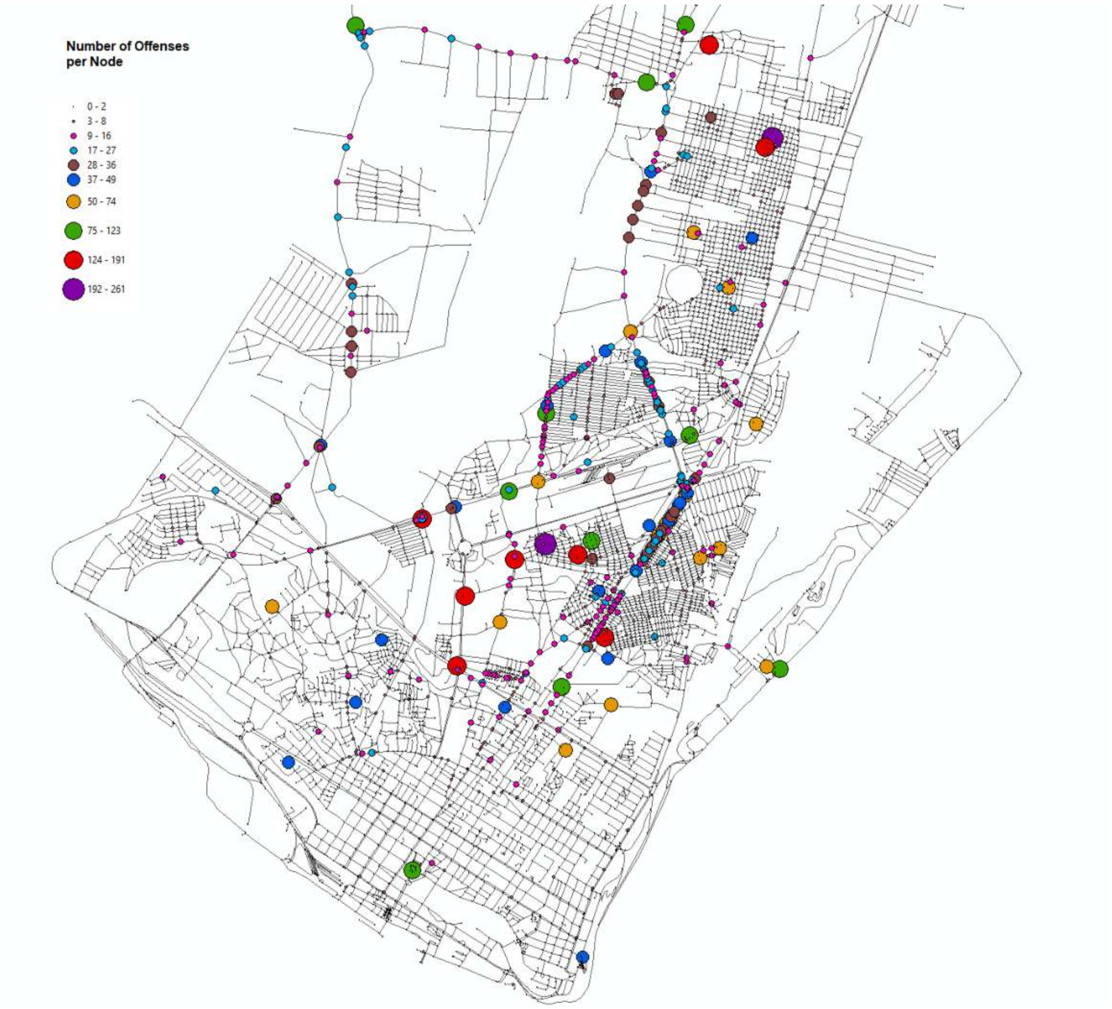


Figure 6.1: Visual output in ArcGIS for the number of offenses per node after the simulation run of the standard model and a time dimension of three simulated years in the city of Maputo.

6.1 General Model Results and Descriptive Analysis

The result data for the standard model was generated during multiple simulation runs with two sets of input parameters. The first set represents default values for all input parameters as presented in Table 4.2 to approximate a realistic setting. This set of input parameters was applied to simulate a time dimension of three years. The simulated time span was chosen by the author as a compromise of simulation duration in real-time and accuracy of output values. The first input data set is used to generate output data for general results and descriptive analysis. General results are first presented as raw data at agent level and furthermore as aggregate data on both agent and society level.

The second set of input parameters corresponds to twelve scenarios, where the `TIME_SAFE` input parameter was varied from 35 % to 90 % in steps of 5 %. The `TIME_SAFE` value is the percentage of the day spent in safe places and it consists of the sum of time spent at activity place home and the activity place work. The simulated time span is one year for each of the scenarios. The resulting data set is analyzed and discussed before it is applied for evaluation of the two hypotheses in the following chapter. The total number of committed crimes, or crime rate, is treated as the endogenous variable, which is explained by the exogenous variables, in particular time spent in unsafe places.

The simulation model generates five output files in CSV format at the end of the simulation. The output files collect individual data from places, citizens, police officers, offenses and an aggregate data report for each simulated day. Descriptive statistics such as mean, median, and range are performed to characterize the results of each of the output files and to compare them.

Daily Report

The daily report output file provides an initial overview of the simulation results. It connects the variables Robberies, Convergences, PreventedByPolice and PreventedByCivilians to each consecutive day of the simulation as shown in Table 6.1. The daily report provides several interesting insights. For example, Robberies and Convergences have a high correlation, as is expected. A higher number of situations where offenders and victims meet, will result in a higher crime rate. In order to prove this suspicion, a correlation matrix is calculated based on the observation of 1096 simulated days with the correlation coefficient $corr(Robberies, Convergences) = 0.758$ indicating a high positive correlation. Under the null hypothesis of no correlation $H_0 : \mu \neq \mu_0$ a T-test results in $t(1093) = 38.4282$ at a significance level of $p < 0.0000$, which means that the null hypothesis of no correlation is refuted. Furthermore, the recorded numbers of the first week of the simulation look similar to those of the last week, which suggests that the crime rate remains constant over the course of the simulation. Similarly, the rates of convergences and prevented crimes do not change significantly. This suspicion is investigated in Table 6.2.

The daily report data also exhibits crime trends, which can be displayed as plot in Figure 6.2 with data from the first 100 simulated days. In order to improve the meaning-

Day	Robberies	Convergences	Prevented By Police	Prevented By Civilians
1	12	17	2	5
2	10	19	0	9
3	12	23	1	10
4	19	28	0	9
5	8	20	0	12
6	11	22	1	10
7	17	23	1	5
...	...	...	...	...
1089	15	21	1	5
1090	11	18	0	7
1091	18	29	0	11
1092	18	29	0	11
1093	14	17	0	3
1094	3	11	0	8
1095	7	16	1	9

Table 6.1: Data from the first and the last week of the report output file.

fulness of the scatter plot of robberies and convergences against days, rolling averages were calculated and plotted as lines. Based on these two lines, the plot clearly indicates a correspondence between Convergences and Robberies. Rises in the purple convergences line have their correspondent rise in the green robberies line. This correlation is an expected behavior, because a convergence, the incident where a capable offender and a viable victim meet in space and time, is a prerequisite for an offense to occur. See Chapter 4.1 for a more comprehensive discussion about the model mechanics.

Calculations from the daily report output data are presented in Table 6.2. The total numbers of robberies (13 208) and convergences (22 063) are reported, as well as the average crime rates per year (4402.67), month (366.89) and day (12.06 and SD = 3.43). The daily crime rate of the simulation is comparable to the crime rate found in empiric data [Uni20], which is normalized to 13.45 street robberies per day and one million citizens. For each convergence, a 60 % chance exists that a successful robbery occurs. The crime prevention rates for police officers (6.9 %) and civilians (93 %) show that random movement of police officers renders them ineffective at crime prevention. Whether the movement of police officers along a systematic route visiting crime hot spots leads to more effective police work is shown in the advanced model results in Chapter 6.2. Finally, mean and standard deviation for average daily crime rate for days 1 to 100, 501 to 600 and 1001 to 1096 indicate that the daily crime rate does not significantly change during the simulation. Based on these numbers it can be concluded that the chosen time dimension of three simulated years is sufficient for meaningful results and a longer time dimension would not have a significant impact on simulation results.

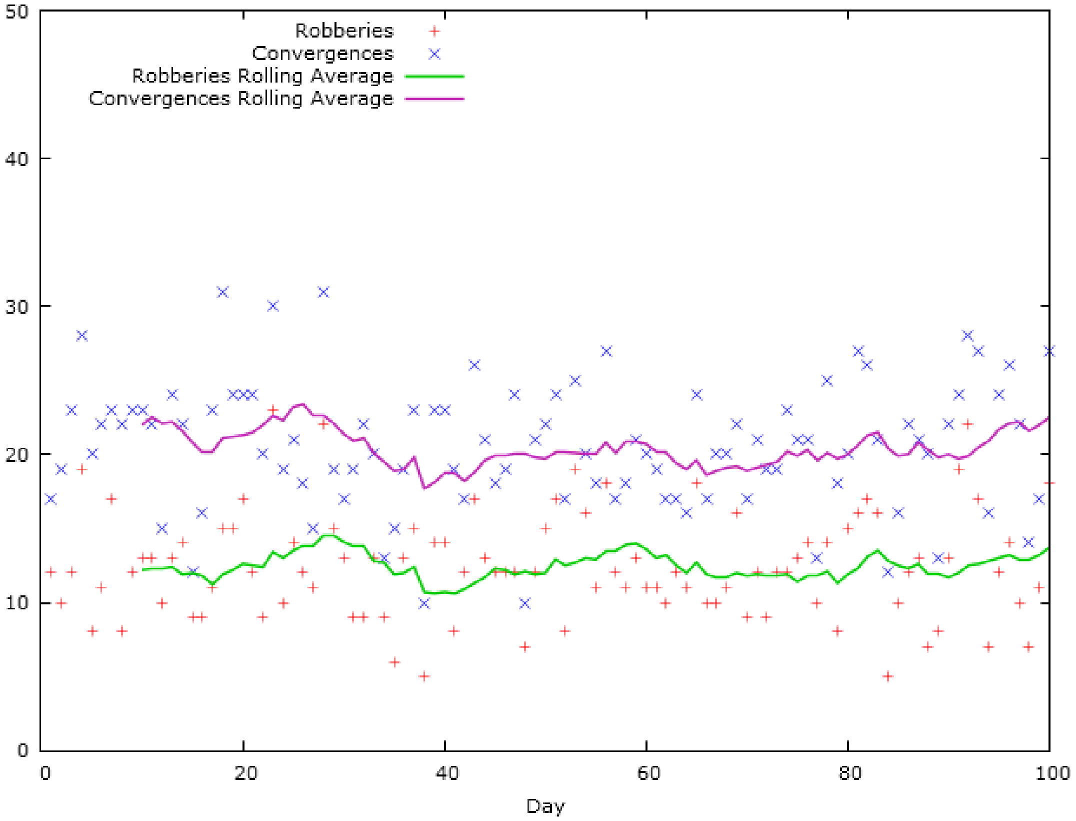


Figure 6.2: Plot of convergences, robberies and their respective rolling averages over the course of the first 100 days of the simulation.

Aggregate Variable	**Values**
Total number of robberies	13 208 (4430 during the first year)
Total number of convergences	22 063 (7389 during the first year)
Average crime rate per year, month and day	4402.67, 366.89, 12.06
Chance of a successful robbery for a convergence	60 %
Total number and percentage robberies prevented by police	629 (6.9 %)
Total number and percentage robberies prevented by civilians	8471 (93 %)
Average daily crime rate for days 1 to 100	12.43 (SD = 3.64)
Average daily crime rate for days 501 to 600	12.40 (SD = 3.53)
Average daily crime rate for days 1001 to 1096	11.75 (SD = 3.49)

Table 6.2: Aggregate values and metrics for daily report output data.

Aggregate Variable	Values	
Number of nodes	7001	
Number of nodes where robberies occurred	1277 (18 %)	
Minimum, maximum and average robberies for a node where robberies occured	1 (*min*), 261 (*max*), 8.07 (*avg*)	
Number and percentage of all nodes visited by citizens	5958 (85 %)	
10 % quantiles for the number of visits per node	10 %	0
	20 %	2190
	30 %	6570
	40 %	16 087.4
	50 %	40 401.5
	60 %	104 315
	70 %	210 282.8
	80 %	484 889.8
	90 %	746 665.9

Table 6.3: Aggregate values and metrics for places output data.

Places Data

Output data for all 7001 places of the simulation is collected in the second output file. The recorded variables are Id, Visits, RobberiesPrevented, Robberies, Convergences, PreventedByCivilians and PreventedByPolice. The results of all 7001 abstract place agents are presented in Table 6.3. From those 7001 nodes, 5958 (85 %) have been visited by citizen or police agents. At least one crime occurred in 1277 of the 7001 nodes (18 %). For those 1277 nodes an average of 8.07 crimes were committed with a minimum of 1 and a maximum of 261. The 10 % quantiles indicate that the number of visits per node are not equally distributed (e.g. 2190 (20 %) and 484 889.8 (80 %)).

The place output data is further applied to calculate of a list of crime hot spots, defined as the ten place agents with the highest number of recorded robberies. Table 6.4 displays the resulting list of crime hot spots for the standard model with a run time of three years. The table includes data for Rank, Node ID, Robberies, Convergences, Prevented by Police and Prevented by Civilians. The Prevented Robberies numbers are the sum of prevented crimes from citizens and police officers. This list is the basis of the hot spot route applied in the advanced model, of which the results are presented in the upcoming Section 6.2.

Citizen Data

The citizen output file collects values of 35 variables for each of the 1000 citizen agents. The variables are Name, InitialWealth, CurrentWealth, Income, CountCommittedRobberies, CountVictimOfRobbery, CriminalPropensity, CumulatedTimeHome, CumulatedTime-

Rank	Node Id	Robberies	Convergences	Prevented by Police	Prevented by Civilians
1	5722	261	436	26	162
2	3419	219	384	0	165
3	5692	191	331	7	136
4	6553	181	257	2	75
5	6020	166	271	0	105
6	1956	165	267	3	100
7	6912	155	231	0	76
8	6451	143	238	12	87
9	3156	139	248	0	109
10	3883	123	239	11	111

Table 6.4: Top ten place agents by number of occurred robberies.

Work, CumulatedTimeActivity1, CumulatedTimeActivity2, CumulatedTimeActivity3, TimePathHomeWork, TimePathWorkAct1, TimePathAct1Act2, TimePathAct2Home, CumulatedTimeTraveling, CumulatedTimeNotSafe, CumulatedTimeUnemployed, TimeHome, TimeWork, TimeActivity1, TimeActivity2, CountRobberiesAct1, CountRobberiesAct2, CountRobberiesAct3, CountRobberiesStreet, CountOpportunitiesAct1, CountOpportunitiesAct2, CountOpportunitiesAct3, CountOpportunitiesStreet, TotalMoneyRobbed, TotalMoneyIncome, InvolvedOpportunities and CurrentTimeUntilNextOffense. Not all of these variables are relevant for general model results, but several variables were used for verification and validation of the model.

The data from the citizen output file serves as basis for further analysis and aggregate measures. The resulting values are presented in Table 6.5. Crime rates are compared between the whole population and different subgroups. The groups of employed and unemployed citizens, the groups of citizens with criminal propensity and law-abiding citizens and groups with a different wealth status are investigated. The simulation resulted an average daily time spent in safe places for all citizens of 68 %. The reader should note that the resulting time spent in unsafe places per day differs from the input `TIME_SAFE` argument due to the unemployment mechanics. Looking at the whole population, 975 out of 1000 citizens (97.5 %) became victims of a crime. The average number of offenses against one citizen agent is 13.55 with a maximum of 352 offenses.

Unemployed citizens visit an unsafe third activity place instead of a safe workplace. The data confirms that those 427 citizens, who were employed for the full duration of the simulation, spent more time in safe places than the 573 citizens, who have been unemployed for at least one period of time. Comparing the percentage of the day spent spent in safe places of employed (75 %) and unemployed citizens (62 %), it is clear that employed citizens spend more time in safe places. As a result, unemployed citizens are expected to be victim of a crime more often than employed citizens. But this is not

Aggregate Variable	Values
Average daily time spent in safe places for all citizens (in ticks and percent of day)	974.34 ticks or 68 %
Average daily time spent in safe places for 427 employed and 573 unemployed citizens	1074.67 ticks or 75 % (employed), 899.58 ticks or 62 % (unemployed)
Number and percentage of citizens who became victim of a crime	975 (97.5 %)
Average number of offenses against one citizen	13.55 (min: 1, max: 352)
Number and percentage of employed victims	415 (of 427) 97.2 %
Number and percentage of unemployed victims	560 (of 573) 97.7 %
Average and total number of offenses against an employed citizen	13.67, 5672
Average and total number of offenses against an unemployed citizen	13.46, 7536
Average daily time spent at home, work, activity place 1, activity place 2, activity place 3, streets	658.77 (46 %), 315.57 (22 %), 116.52 (8.1 %), 65.931 (4.6 %), 101.60 (7.1 %), 181.61 (13 %)
Total count and percentage of offenses at activity place 1, activity place 2, activity place 3, streets	2818 (21 %), 1803 (14 %), 430 (3.3 %), 8157 (62 %)
Number and percentage of citizens with a criminal propensity	87 (8.7 %)
Number of offenses against citizens with a criminal propensity	939 (10.79 per citizen)
Number of offenses against citizens with no criminal propensity	12 269 (13.44 per citizen)
Average number of times a citizen with criminal propensity committed an offense	151.82 (min: 1, max: 396)
Average cumulated amount of money robbed per offender	378.47 (min: 26, max: 7977)
Comparison of the 10 %, 50 % and 90 % quantiles of initial wealth	10, 450.5, 1662
Comparison of the 10 %, 50 % and 90 % quantiles of final wealth	500.3, 1236.5, 2374.4
Total number of offenses against citizens from the most wealthy 10 %, to the least wealthy 10 %	1277, 1500, 1105, 968, 978, 1483, 1718, 1502, 1279, 1398

Table 6.5: Aggregate values and metrics for citizen output data.

the case as deeper analysis reveals. Considering the employment status, 415 employed and 560 unemployed citizens became victims of a crime, which is 97.2 % of the group of employed and 97.7 % of the group of unemployed citizens. The total number of offenses against employed citizens is 5672 and against unemployed is 7536. In average numbers, one employed citizen becomes a victim of a crime 13.67 times and an unemployed citizen 13.46 times.

The results also show how much time a citizen spends at each activity space on average. The investigated places are home (46 %), work (22 %), activity place one (8.1 %), activity place two (4.6 %), activity place three (7.1 %) and streets (13 %). Further, the number of offenses are counted for the victim's activity spaces. The results show that 2818 offenses took place at activity space one (21 %), 1803 at activity space two (14 %), 430 at activity space three (3.3 %), and finally 8157 occurred at street nodes (62 %).

A similar comparison can be applied to investigate the differences between the group of citizens with a criminal propensity (87 citizens) and the group with no such propensity (913 citizens). The number of offenses against citizens with a criminal propensity is 939, or 10.79 per citizen. The number of offenses against citizens with no criminal propensity is 12 269, or 13.44 per capita. On average, the number of times a citizen with criminal propensity committed an offense is 151.82 with a maximum of 396 offenses and the average offender robbed a sum of 378.47 money units with a minimum of 26 and a maximum of 7977 during the three simulated years.

To investigate the relevance of the wealth status of victims, the whole population is divided in ten classes by their final wealth attribute. Wealth plays an important role during the decision of a robbery, because the offender must have a higher wealth status than the victim, or else the robbery cannot occur. The poorest 10 % of the population owned an average amount of 500.30 money units at the end of the simulation and the wealthiest 10 % owned 2374.40 money units. These are the total number of offenses against citizens from the most wealthy 10 %, to the least wealthy 10 % 1277, 1500, 1105, 968, 978, 1483, 1718, 1502, 1279, 1398.

Police Officers Data

The output file collecting results from all 200 police officer agents features four variables: Id, NumberOfDistinctNodes, RobberiesPrevented and IsHotspotCop. The evaluation of the output data for 200 police officer agents as shown in Table 6.6 exposes some inherent deficiencies of the initial placement and undirected, random movement of police officers in the standard model. 629 crimes were prevented by police officer agents out of 22 063 opportunities. The rate of crime prevention of all police officers (2.9 %) is sub-par compared to the rate of civilian bystanders (38 %). Considering that 42 out of 200 police officers did not prevent a single crime, deeper analysis is required to find the cause of this phenomenon. The solution is connected to the number of distinct nodes each police officer visits during the simulation. The distinct nodes variable counts the number of nodes a police officer visits during his random movement around the street network.

Aggregate Variable	**Values**
Total number of police officer agents	200
Total number of crimes prevented	629
Comparison of police and civilian prevention rates	2.9 % (police), 38 % (civilians)
Efficiency of police officers (in prevented offenses per officer)	3.15
Number of police officers who did not prevent a single crime	42 (21 %)
Average and maximum number of distinct nodes visited by police officers	1120.65 (*avg*), 2225 (*max*)
Number of police officers with $DistinctNodes <$ 100 (locked in small circle)	75 (38 %)
Total number of crimes prevented by police officers with $DistinctNodes < 100$ and ≥ 100	202 (32 %), 427 (68 %)
Efficiency of police officers with $DistinctNodes <$ 100 and $\geq$ 100 (in prevented offenses per officer)	2.69, 3.42

Table 6.6: Aggregate values and metrics for police output data.

Aggregate Variable	**Values**	%
Robberies at victim location Path from Home to Work	2509	19
Robberies at victim location Activity 3	429	3
Robberies at victim location Path from Work to Activity 1	2761	21
Robberies at victim location Activity 1	2818	21
Robberies at victim location Path from Activity 1 to Activity 2	2888	22
Robberies at victim location Activity 2	1803	14
Total	13 208	100.00

Table 6.7: Aggregate values and metrics for offenses output data.

The average number of distinct nodes is 1120.65 (2225 *max*), but the number of police officers with a very low number of distinct nodes is more relevant. Calculations show that 75 police officers are trapped in a small circle of less than 100 distinct nodes. These 75 officers only prevented 202 crimes (32 % efficiency), while their colleagues with 100 and more distinct nodes managed to prevent 427 crimes (68 % efficiency). An attempt to improve the effectiveness of the police force is implemented in the advanced model (see Section 6.2 for results), where one half of the police officers patrol a circular route involving the top ten crime hot spots from the standard model.

Scenario	TIME_SAFE Parameter	Average Time in Unsafe Places	Convergences	Offenses
1	35 %	1034.27	14 403	8690
2	40 %	964.81	13 341	8060
3	45 %	893.56	12 484	7465
4	50 %	821.16	11 996	7309
5	55 %	749.42	11 032	6635
6	60 %	677.17	10 315	6139
7	65 %	605.70	9180	5498
8	70 %	533.54	8594	5085
9	75 %	461.67	7483	4451
10	80 %	388.99	6408	3732
11	85 %	318.70	5120	2969
12	90 %	253.24	3748	2179

Table 6.8: Results for twelve scenarios of different TIME_SAFE input parameters for one simulated year.

Offenses Data

Data from offenses was collected in order to investigate the locations where offenses occurred as shown in Table 6.7. The street location was split in three sections according to a citizen's daily route. The first section starts at the home location and ends at the work location. The second section leads from work to activity one and the last section leads from activity one to activity two. For these six locations the number of offenses are 2509 for the first section of the path, 429 for activity three, 2761 for the second path, 2818 for activity one, 2888 for the last path and 1803 for activity three.

The collected data from offenses also includes the timestamp of every offense. This facilitates the investigation of the temporal distribution of offenses as visualized in Figure 6.3. From 0:00 to 9:00, the citizens stay at home safely and the number of offenses remains zero. Then, citizens start along their path to work, or activity space 3, which explains the first spike in the graph at around 12:00. The number of offenses falls rapidly after the midday peak between until the minimum is reached at 16:00, when most citizen agents reside at their work node. After that, they move along the street network and stay at recreational activity places one and two, while being at risk of robbery for the whole time. The offense per hour count reaches its maximum at 23:00.

Scenarios for the Varied TIME_SAFE Input Parameter

The second set of input parameters was applied to to twelve simulation scenarios, where the TIME_SAFE input parameter was varied from 35 % to 90 % in steps of 5 %. The results of the twelve simulation scenarios are summarized in Table 6.8. The variation of the TIME_SAFE input parameter, which determined the amount of time spent at home

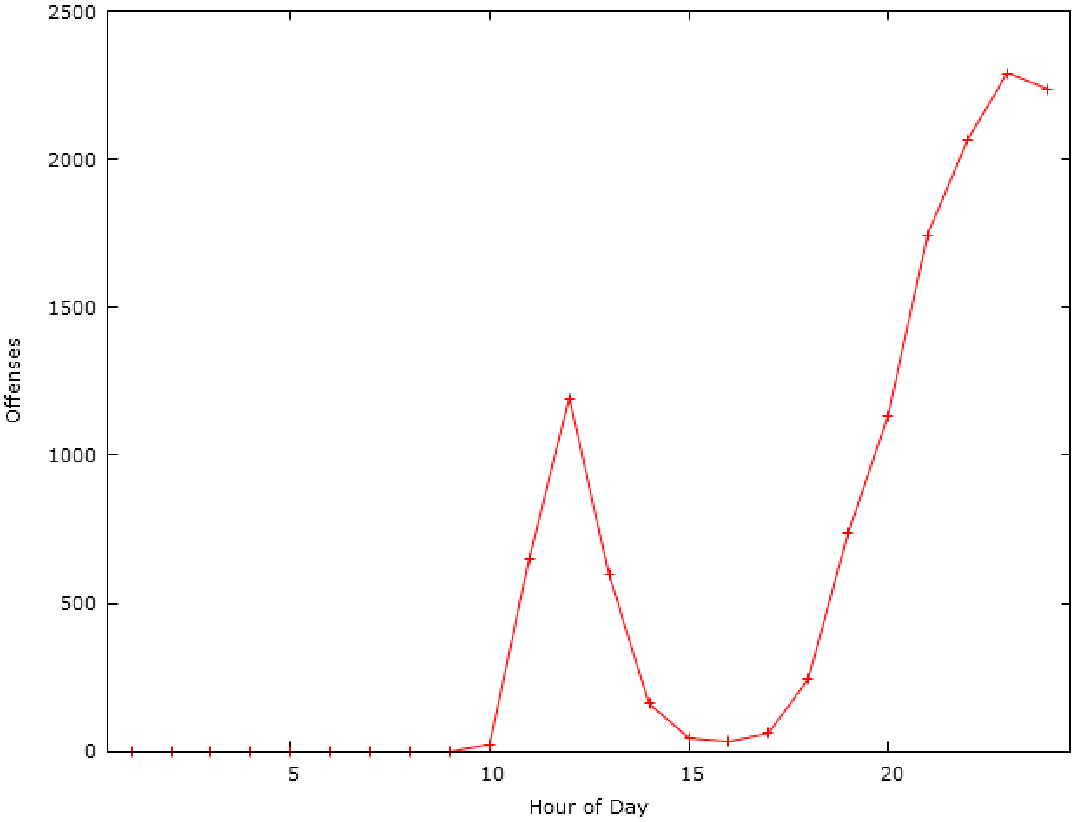


Figure 6.3: Plot of the number of offenses per hour of the day.

and work, resulted in a significant change in the resulting average time in unsafe places. In scenario one, the average citizen spent 1034.27 ticks (17 hours and 14 minutes) of a day at risk of being robbed and in scenario twelve the average citizen was at risk for 253.24 ticks (4h 13min). The differences of the duration where citizens were at risk resulted in significant differences of convergences and offenses. The number of convergences started at 14 403 for scenario one and decreased from each scenario until 3748 convergences were reported for scenario twelve. Similarly, the number of offenses decreased from 8690 in scenario one to 2179 in scenario twelve. The ramifications of the outcomes of the twelve scenarios for the hypothesis evaluation are discussed in this chapter in Section 6.3.

6.2 Advanced Model Results

In the advanced model, one half of the police officers move along a predefined route, connecting the identified crime hot spots from the standard model, while the other half still move randomly. All other agent rules and input parameters, especially the RNG seed, remain unchanged. The societal-level results of the simulation run of the advanced model with a temporal dimension of three years are summarized in Table 6.9. The table

Variable	**Advanced Model**	**Standard Model**
Offenses	11 105	13 208
Convergences	22 617	22 063
Prevented by Civilians	8793	8471
Prevented by Police	4453	629

Table 6.9: General results of the advanced model and comparison to the standard model.

Node Id	**Offenses**	**Convergences**	**Prevented by Police**	**Prevented by Civilians**
5722	5	430	423	151
3419	6	459	449	198
5692	2	364	358	142
6553	5	326	320	128
6020	5	342	335	108
1956	2	289	286	109
6912	3	253	250	89
6451	136	224	6	83
3883	81	185	50	79
3156	137	281	34	120

Table 6.10: The top ten place agents by number of offenses from the standard model after a simulation run of the advanced model.

also contains the corresponding results from the standard model. The number of offenses decreased from 13 208 (standard) to 11 105 (advanced), a decline of 2103 offenses or 16 %. The recorded convergences and the prevented offenses by civilians only changed slightly (2.5 % and 3.8 % respectively). This was expected because both mechanics are not directly affected by the movement patterns of police officers. The prevented offenses by police officers increased from 629 (standard) to 4453 (advanced), an increase of 3824 prevented offenses or 608 %.

A list of the node agents with the highest number of offenses has been created from standard model results and shown in Table 6.4. For further analysis of the proposed research question, it is necessary to investigate the status of those ten nodes based on the outcome of the advanced model. The resulting data is summarized and displayed in Table 6.10. For seven of the ten nodes, the crime rates dropped significantly to numbers between two and six offenses. This decrease of crime rates comes with a parallel increase of prevented offenses by police officers. The last three nodes do not follow the trend. Their numbers of offenses decreased for two of the three nodes, but did not change in the same order of magnitude of other seven places. Similarly, the crime prevention by police officers improved, but by a lower factor. Further investigation is necessary to determine the reason for the divergent behavior of nodes 6451, 3883 and 3156.

6.3 Hypothesis Evaluation

The first research question

> *RQ 1: Is the routine activity theory, which assumes that crime rates decrease when there is more time spent at safe places, valid for an agent-based street crime model representing Maputo City?*

is investigated based on the data in Table 6.8. The column „TIME_SAFE Parameter“ contains the values for the independent variable, which was varied for twelve simulation runs from 35 % to 90 % in steps of 5 %. The column „Offenses“ contains the values for the corresponding crime rates for each of the twelve scenarios as dependent variable. In general, the research question is to investigate the correlation between two variables. The variables are interval scaled and the correlation is assumed to be linear. There is one explanatory variable, TIME_SAFE, so a simple linear regression model is used. The linear regression model describes the correlation between two variables as a linear function. The dependent variable y is a function of the independent variable x_i:

$$y = f(x_i)$$

The regression model consists of the observed values of the dependent variable *offenses*, the observed values of the independent variable *timesafe*, the error term ε_i of the sample, the intercept β_0 and the regression coefficient β_1.

$$\text{offenses} = \beta_0 + \beta_1 * \text{timesafe} + \varepsilon_i$$

Prior to analyzing the results of the regression analysis, the assumptions are evaluated. The investigated Gauss–Markov assumptions are linearity of the correlation, linearity of the coefficients, the mean of zero of the error term, homoscedasticity, independence and the normality of errors. For visual proof of the linearity of the correlation, a scatter plot between the dependent and the independent variable is presented as Figure 6.5. It clearly shows a negative correlation between the varied TIME_SAFE parameter and offenses. Gauss–Markov assumption two is fulfilled, because the estimated parameters are linear as shown in the regression model earlier. The third Gauss–Markov assumption postulates that the expected value of the error term is zero for all observations: $E(\varepsilon_i) = 0$. Looking at the residuals of the regression, the mean is 3.41×10^{-13}, which is close to zero. The fourth Gauss–Markov assumption expects the error term to have constant variance for every value of the independent variable $Var(\varepsilon_i) = \sigma^2$ (homoscedasticity). This can be verified by investigating the scatter plot the residuals in Figure 6.4. The fifth Gauss–Markov assumption requires the error term to be independently distributed and not correlated: $Cov(\varepsilon_i, \varepsilon_j) = E(\varepsilon_i \varepsilon_j) = 0, i \neq j$. This assumption is also verified in

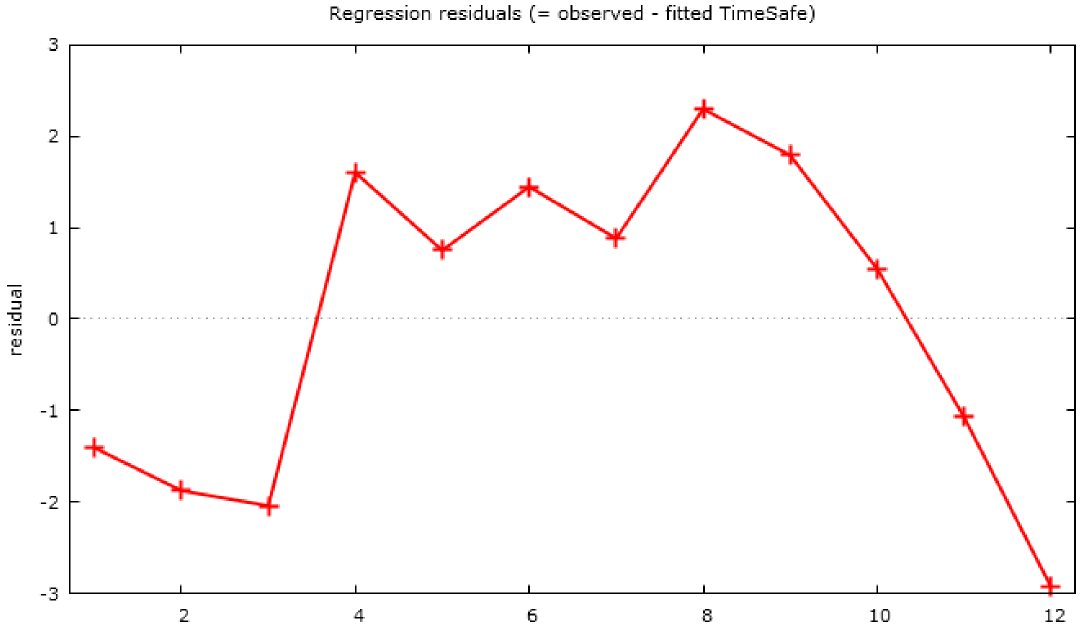


Figure 6.4: Plot of the regression residuals.

	Coefficient	Std. Error	t-ratio	p-value
const	12815.1	229.964	55.73	0.0000
TimeSafe	−114.092	3.54666	−32.17	0.0000

Mean dependent var	5684.333	S.D. dependent var	2066.742
Sum squared resid	449692.2	S.E. of regression	212.0595
R^2	0.990429	Adjusted R^2	0.989472
$F(1, 10)$	1034.840	P-value(F)	1.98×10^{-11}

Table 6.11: OLS, using observations 1–12. Dependent variable: Offenses.

Figure 6.4. The last assumption that is tested requires that the error term is normally distributed. This requirement is verified by a normal probability plot.

After ensuring that all requirements are fulfilled, the regression model and the coefficients are tested for significance. To make sure that the regression model is significant, an F-test is performed. The F-test determines whether the prediction of the dependent variable improves by adding the independent variable. The F-value for this model is shown in the regression results in Table 6.11. The results of the F-test with $F(1, 10) = 1034.840$, $p = 1.98 \times 10^{-11}$ indicate that the model is significant.

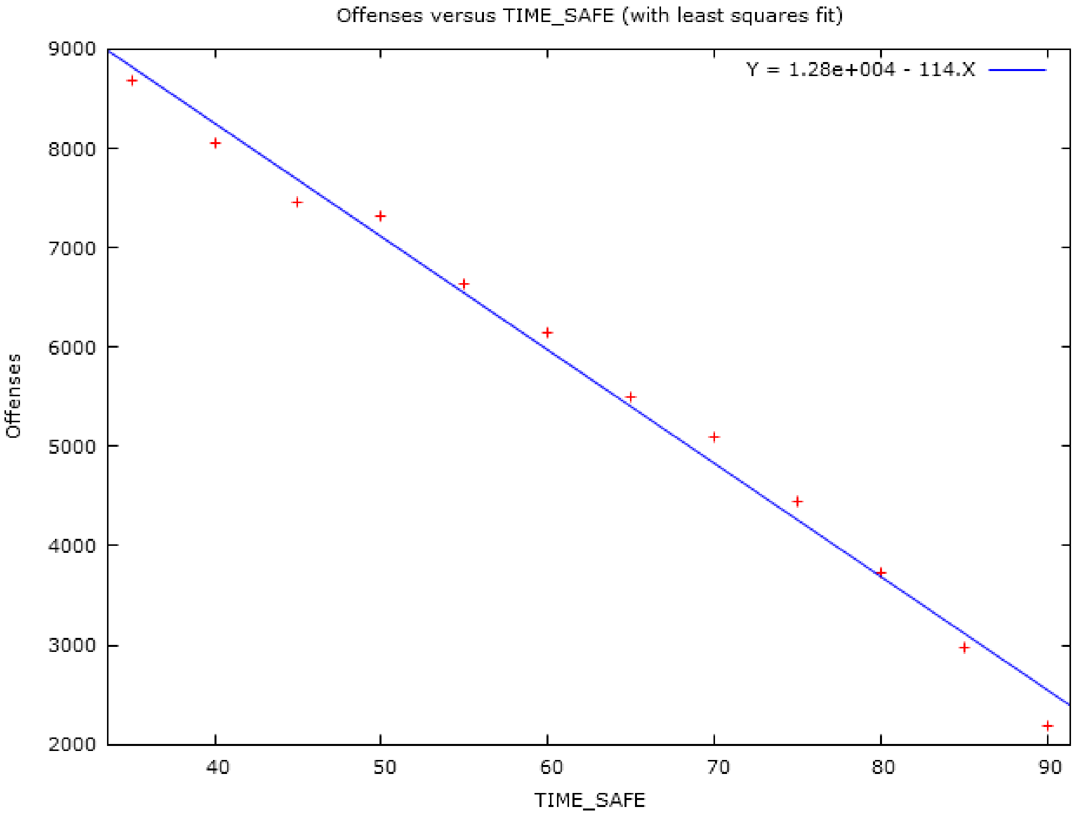


Figure 6.5: Plot of the resulting offenses versus the `TIME_SAFE` input variable with least squares fit.

The coefficient estimates as listed in Table 6.11 indicate that for every one percent increase of the `TIME_SAFE` parameter, the offenses decrease by 114.09. In order to verify that the coefficients of the regression are significant, a t-test is performed for every coefficient. The results of the t-tests are displayed in Table 6.11. The results indicate significance for both the constant ($t = 55.73$, $p < .001$) and the coefficient *TimeSafe* ($t = -32.17$, $p < .001$). The significance of the constant implies that the regression line does not intersect the coordinate system origin. The significance of the coefficient *TimeSafe* implies that the regression coefficient of *TimeSafe* does not equal zero and *TimeSafe* has a significant impact on the number of offenses. The result is the following regression line:

$$\text{offenses} = 12\,815.1 - 114.092 * \text{timesafe}$$

In order to determine the goodness-of-fit for the linear model, the R-squared statistic is investigated. R-squared is a statistical measure of how close the observed data is to the estimated regression line. It shows how well the estimated model fits to the observed data. R-squared assumes values between 0 % and 100 %. 0 % implies that the model explains none of the variability of the response data around its mean and 100 % explains all the variability of the response data around its mean. The R-squared value is influenced by the number of independent variables, so the adjusted R-squared is taken into consideration. In the present linear regression model the adjusted R-squared value is 0.989, which implies that close to 98.9 % of the dependent variable *offenses* is explained by the TIME_SAFE parameter. It can be concluded that the percentage of the day spent in safe places does have an influence on the number of committed offenses.

The second research question

> *RQ 2: Are law enforcement officers, who patrol along the crime hot spots of the standard model, significantly reducing the crime rates in the advanced model? To what extent does relocation of crime to new clusters occur?*

is examined based the daily report simulation results and on data listed in Table 6.9. The advanced model introduced a predefined route, where one half of the police officers moved along, while the other half still followed a random movement like in the standard model. The number of offenses decreased from 13 208 (standard model) to 11 105 (advanced model), a decline of 2103 offenses or −16 %. In order to determine whether there is a significant difference between the crime rates of the standard and the advanced model, a dependent samples t-test is used. Under the null hypothesis that the difference of means equals zero, the means are calculated for 1095 observations of daily crime rates for both the standard (mean = 12.0621, SD = 3.42666) and the advanced model (mean = 10.1416, SD = 2.99131). Then, the test statistic is computed: $t(2188) = 13.9718$. The test statistic is compared to the critical value of the t-distribution of the given degrees of freedom (1094) at a significance level of 0.05. The critical value is 1.64625. From $13.9718 > 1.64625$ follows that the null hypotheses must be refuted and the means are significantly different.

The second part of research question two can be answered by comparing the place agents with the highest crime rates of the advanced model to those of the standard model. For example, node id 2339 registered 188 offenses in the advanced model and zero offenses in the standard model. This node represents a new crime hot spot that emerged in the advanced model. Table 6.9 shows that the crimes prevented by the police increased from 629 to 4453 (3824), while the overall offenses decreased only by 2103. Hence, for 1721 of the 3824 additional preventions, an offense took place at another node away from the hot spot route. The relocation of crime accounts for 45 % of prevented offenses.

CHAPTER 7

Discussion

This chapter discusses the novelty of the approach, the results of this work and performs a comparison of the findings with the defined goals. In the first section, the validity of the model is discussed. According to the presented methodology in the introduction, the validity of the selected approach is shown. The correct application of the scientific methodology is the foundation for the validity of the results. In the second section, the simulation results are assessed. The results are compared to empirical data. In the next step, research questions are evaluated based on the simulation results. In the final section, limitations of the research are discussed.

As described in Section 1.1 this work is placed in the context of the routine activity theory, a sociological theory published by Cohen and Felson in 1979. The researchers investigated the paradoxical phenomenon of increasing crime rates during a time of economic growth and prosperity. Their approach to finding an explanation is the shift of activities from the households to work and leisure activities. While the authors provided empirical evidence of household activities in relation to crime statistics, the validity of this theory is still debated. The acquisition of individual-level data of everyday life activities is difficult and often not possible due to privacy laws and regulations. The challenge of collecting individual-level data can be overcome by an agent-based modeling and simulation approach.

This work applies modeling and simulation as a scientific method for evaluating the routine activity theory. The type of criminal acts under study are direct-contact predatory violations, or street robbery. Other types of crime are not modeled. The first research question concerns the hypothesis that crime rates decrease when there is more time spent at safe places, while other factors remain constant. The simulation generated data that supports this hypothesis with significance. The second research question concerns the effectiveness of a policing strategy. The results confirm a significant decrease of the crime rate when this policing strategy is applied.

In order to achieve the evaluation of the research questions, intermediate results were produced. The first intermediate result is „The Street Crime Model“ (see Chapter 4). In this model, citizen agents perform their everyday routines including home, work and two leisure activities. Outside of home and work, citizens are at risk of being robbed by other citizens with a criminal propensity. The model describes the three elements of crime: the motivated offender, the suitable target and the absence of a capable guardian. The second intermediate result of this work is the implementation of the model (see Chapter 5) in a computer language. The implemented model is written in Python and uses the simulation software Agent Analyst. The simulated environment is based on a real street map of Maputo City.

Various simulation runs were performed in order to generate result data at the individual and at the aggregate level. The results of the simulation and the evaluation of the hypotheses are presented in detail in Chapter 6.

The findings of this work are applicable to cities that are comparable to the city of Maputo. The applicability of results to other cities depends on the main demographic attributes that serve as parameters of this model. The parameters are population, unemployment rate, income, wealth and crime rate of street robberies. The city street layout is also a factor to be considered, because it influences the spatial distribution of crime and the convergence of offender and victim. A higher similarity in terms of the presented parameters will lead to stronger applicability of the results.

This work builds on existing contributions (see Chapter 3 Context and Related Work) to the topics of routine activity theory and agent-based simulation and investigates the subject using an improved model in terms of agent decision making, agent movement and modeling of the environment. Two related works of Elizabeth Groff are of note. The findings of Groff's first and simplified routine activity model [Gro07a] with Seattle as object of study also shows strong support for the proposition of the routine activity theory, which states that the number of crimes increases when people spend more time away from home. However, this simple model does not realize routine activity spaces for citizen agents, which are a core component of the routine activity theory. In the author's view, the results are limited in terms of validating the routine activity theory. Groff's second model [Gro06] adds a spatial and temporal schedule for citizen agents, but the results do not support the main proposition of the routine activity theory in contrast to the findings of this work. The improved model of this thesis should provide more trust with regards to the evaluation of the results. Another novelty of this approach is that crime hot spots and different policing strategies are evaluated relating to the routine activity theory. The evaluation of the routine activity theory combined with an African capital city as an object of study is a new approach and results in new scientific findings.

7.1 Validity of the Model

The methodological approach of this work is presented in Chapter 1.4. The stages of simulation-based research (see Chapter 2.2.2) were identified from the available literature and the approach of this work was designed to follow this established methodology. According to the planned approach, a model was created from the theoretical basis of GIS, ABS and the routine activity theory. The goal was to deduct a simplified representation of the real world as the Street Crime Model. During the implementation step, the simulation model was created from the specifications provided by the theoretical model. The calibration of the model was conducted by selecting input parameters of the simulation from empiric data. Where no data was available, assumptions were made and argued.

The set of certification measures to ensure the validity of the model mainly consists of the verification and validation processes and the sensitivity analysis. The performed steps of verification measures are documented in Chapter 5.5. The goal of the verification process was to certify that the implemented model is working correctly. A high degree of software correctness can be asserted by the application of established techniques and procedures. The computer program was systematically tested against this specification. Structured code walkthroughs were performed. A systematic evaluation of logging outputs asserted the correctness of the implemented methods.

In the validation process, the task was to ensure that the model outcome is useful to understand the real phenomenon as described in the problem statement. The validation activities of this work are documented in Chapter 5.6. According to the presented methodology the internal and the external validity of the model were tested. The internal validity was evaluated by comparing the key aspect of the conceptual model, that crime rates increase when the activities of people shift away from their homes, to the behavior of the implemented model. The results reproduced the behavior of the target as displayed in Table 6.8. The external validity was achieved by asserting that the simulation reproduces empirical data. The reason is that when a simulation reproduces empirical data, the validity of the model increases. Two central variables were chosen for the validation process. The simulation results, as documented in Chapter 6.1, show that the average daily crime rate of the simulation (12.06) is comparable to the crime rate found in empiric data (13.45). The `TIME_SAFE` value represents the percentage of the day spent in safe places by a citizen agent. For this metric, the empirical data (68 %) agrees with the observed result data (68 %).

In order to certify the robustness of the model, a sensitivity analysis was performed as described in Chapter 5.7. The four exogenous variables, for which the sensitivity analysis was performed, are unemployment rate, minimum time between offenses, criminal propensity and random number seed. The results provide a strong indicator for the robustness of the model. The results of the verification and validation process and the sensitivity analysis ascertain the validity of the simulation results. Based on the performed measures, one can conclude that the simulation model is a valid model of the target system. The key behavior of the target is reproduced and the results reproduce empirical data. The influence of the independent variables on the dependent variable was assessed.

7.2 Assessment of the Simulation Model Results

In this section, the general results of the simulation are assessed at the individual and at the aggregate level. A high positive correlation between convergences and robberies was determined. This emphasizes one aspect of the routine activity theory, which states that crime is a product of opportunity as the convergence of offender, target and the absence of a guardian. Based on the aggregate data, the average number of robberies per day was constant over the course of the simulation. This means that the chosen time dimension of three simulated years was sufficient for obtaining relevant results.

The analysis of place output data shows that crime incidents were not distributed equally across all 7001 places of the Maputo City street network. Clusters of crime were developed during the simulation. The node with the highest number of occurred robberies counted more than twice as many robberies (261) than place number ten on the hot spot list (123). The clustering of crime as shown in Figure 6.1 is a result of the daily routines of the citizen agents. Higher-level streets were used more frequently than back streets and hot spots tendencially emerged on those „highways“.

The simulation resulted an average daily time spent in safe places for all citizens of 68 %. Safe places are the home and work activity spaces as defined in Section 4.2. Cohen and Felson [CF79] investigated the average time citizens spend at home and work based on empiric data and showed that the time safe value is 68 %.

Several attributes of individual-level data of citizens were investigated: percentage of the day spent in safe places, employment status, criminal propensity and wealth. Comparing the percentage of the day spent in safe places of employed (75 %) and unemployed citizens (62 %), it is clear that employed citizens spend more time in safe places. The reason lies within the design of the model, where the work place is safe, as opposed to the third activity place of unemployed citizens as defined in Section 4.2. Comparing the average daily time spent at the three activity places to the crime rates of those places, it is evident that streets are the places with the highest crime rate (62 %). The reason for this emergent phenomenon cannot be identified with certainty, but it might be a consequence of the higher frequency of citizen visits. Regarding the two groups of citizen agents with a criminal propensity and with no criminal propensity, the results showed that criminals were not at higher risk of being robbed than citizens without a criminal career. The wealth status was also investigated. An analysis of ten wealth classes did not indicate that wealthy citizens have a significant higher chance to become a victim of a crime than poor citizens.

The crime prevention rates for police officers (6.9 % of all prevented robberies) and civilians (93 %) showed that random movement of police officers renders them ineffective at crime prevention. Deeper analysis of the random movement of police officers revealed that 75 out of 200 police officers were trapped in small circles of less than 100 distinct nodes, which significantly impaired their effectiveness at crime prevention. This unintended behavior is caused by the quality of the street network and the random placement of police officers.

The investigation of the temporal distribution of offenses resulted in two peaks of high criminal activity during a day. The first occurred at noon, when the majority of citizens traveled from home to work and the second occurred in the evening after the work activity with a maximum at 23:00.

An analysis of the advanced model was conducted with aggregate and place-level results. The number of offenses decreased by 16 %. The prevented offenses by police officers increased by 608 %. This means that the application of a systematic movement pattern for one half of the police agents improved their effectiveness at crime prevention by a factor of seven. For seven of the ten hot spots of the standard model, the crime rates dropped significantly, parallel with an increase of prevented offenses by police officers. The results proved the success of the hot spot policing strategy. It can be reasoned that a systematic and focused presence of police officers at hot spots lowers the crime rate of those places.

7.3 Research Questions and Findings

The first research question

> *RQ 1: Is the routine activity theory, which assumes that crime rates decrease when there is more time spent at safe places, valid for an agent-based street crime model representing Maputo City?*

was answered by investigating the twelve simulation scenarios where the independent variable `TimeSafe` was varied from 35 % to 90 % in steps of 5 % in order to study the resulting dependent variable `CrimeRate`. All other parameters remained unchanged. The variation of the `TimeSafe` input parameter resulted in a significant change of the resulting average time in unsafe places. Likewise, the number of offenses decreased from 8690 in scenario one to 2179 in scenario twelve. The research question requires an investigation of the correlation between two variables `TimeSafe` and `CrimeRate`. There is one explanatory variable, `TimeSafe`, so a simple linear regression model was used. The regression is demonstrated in detail in Chapter 6.3. An F-test was performed with the result that the regression model is significant. A t-test was performed for every coefficient with the result that the parameter `TimeSafe` has a significant impact on the resulting `CrimeRate`. According to the R-squared value, 98.9 % of the dependent variable `CrimeRate` is explained by the `TimeSafe` parameter. It can be concluded that the percentage of the day spent in safe places does have an influence on the number of committed offenses. The evaluation of the simulation data shows that the first research question can be answered positively. The routine activity theory, which assumes that crime rates decrease when there is more time spent at safe places, is valid for the presented agent-based street crime model of Maputo City. This work showed that the major factor for crime in this model is opportunity. An increase in opportunities causes an increase in criminal acts, even though the number of potential criminals and other aspects remain unchanged.

The second research question

> *RQ 2: Are law enforcement officers, who patrol along the crime hot spots of the standard model, significantly reducing the crime rates in the advanced model? To what extent does relocation of crime to new clusters occur?*

was evaluated based on the results of the advanced model. The number of offenses decreased by 16 %. In order to determine whether this is a significant difference, a dependent samples t-test was used. Under the null hypothesis that the difference of means equals zero, the means were calculated for 1095 observations of daily crime rates for both the standard (mean = 12.0621, SD = 3.42666) and the advanced model (mean = 10.1416, SD = 2.99131). The t-test resulted with the findings that the null hypothesis was refuted and the means were significantly different. Hence, the first part of the research question two is confirmed to be valid. The introduced policing strategy of patrolling along the crime hot spots of the standard model does indeed significantly reduce crime rates in the advanced model.

The second part of research question two was analyzed by a comparison of the place agents with the crime rates of both models. The findings reveal that new hot spots emerged in the advanced model. This observed relocation of crime accounts for 45 % of prevented offenses. The second part of research question two is thus also answered. The extent of the observed relocation of criminal acts to new clusters is 45 % of prevented offenses, or 1721 in absolute numbers. This means that the introduced policing strategy could only eliminate crime to a certain extent (−16 %), because the locations of crime moved to other clusters. This phenomenon is also called *crime displacement* and it is a recognized research area in criminology.

7.4 Limitations of the Research

This section summarizes limitations of the research and the used methods. It also aims to provide some grounds for improvements for future work in this area. First of all, the implications of the use of the research method agent-based modeling and simulation need to be regarded. A simulation model is always a simplified model of the reality and as such it does not perfectly imitate the behavior of the reality. The goal of modeling is to design the model as simple as possible, but with enough complexity, so that it works well enough for its purpose. The use of ABM in social science is challenging, because the investigated subjects are humans. Human behavior is approximated by a set of rules. The type of crime „robbery“ was selected deliberately, because it involves some extent of planned action from the offender, as opposed to expressive crime, which often originates from affect. But humans do not always decide rationally, which is the main criticism of the rational choice theory. Techniques to imitate human behavior with an increased complexity have been established as described in Chapter 2.2.5. But the focus and the scope of this work were not to build complex agents, but to investigate emergent phenomenons from simple rules.

Two apparent simplifications concern the citizen agents. While the real population of Maputo City was 1.104 Million in 2020, only 1000 citizen agents were simulated due to limitations of the simulation software and the used computer hardware. Still, 1000 subjects are enough to obtain statistically relevant conclusions. The other simplification concerns the static activity spaces of the citizens. While each of the 1000 agents features its own unique set of activity spaces, the number of spaces is limited to four and all citizens travel along their same route every day with no variety. An improved simulation model would not require the computation of all citizen activity spaces in advance, but citizens would rather decide in real time where to go next or what to do next based on their set of rules and attributes. Other simplifications concern the attributes sex, age and education of individuals.

One relevant limitation concerns the geography and the socio-economic attributes of places. The street network of Maputo City did not contain additional data concerning population density, distribution of income and wealth, or land use. Different means of transportation were also omitted.

The impact of model parameters was assessed by the sensitivity analysis. The high number of police officer agents (200) compared to citizen agents (1000) is due to the random movement of police agents. A lower number would result in almost no prevented offenses. In the advanced model, the police force applies a policing strategy with a higher success of prevented offenses. In this case, a lower and more realistic ratio of police officers to citizens would be feasible.

Concerning the development environment, it must be concluded that Agent Analyst as middleware between ABMS and GIS software does not support important concepts of modern software engineering such as unit tests. In terms of IDE features, the editor of Agent Analyst works as most basic text editors. Agent Analyst is based on Repast Py, which is discontinued and limited in terms of features a modern IDE offers. Repast Simphony 2.8.0, released in October 2020, is the most recent version of the simulation platform and it offers modeling and simulation of spatial models out-of-the box. The programming environment is based on Eclipse and it supports unit tests.

CHAPTER 8

Conclusion and Outlook

This chapter provides a summary of the work done in this thesis. In order to outline the applied methodology, the research goal, the scientific approach and the performed steps are elaborated. Afterwards, the developed artifact, the produced results and the evaluation of the research questions are presented. Finally, the relevance of the work is demonstrated. The second section formulates ideas for improvement and outlines promising approaches for future work in this area. Possible research areas are highlighted and other applications of the proposed solution are discussed.

8.1 Summary of Results and Hypothesis Testing

The routine activity theory was developed by the sociologists Cohen and Felson in 1979. The goal was to explain rising crime rates in a period of economic and social improvements. They reasoned that a shift in the routine activities of citizens away from their homes facilitated a higher number of opportunities for crime, especially street robbery. Cohen and Felson further elaborated that a crime such as robbery requires three elements to occur: a motivated offender, a suitable target and the absence of a guardian. The empirical validation of the routine activity theory is difficult, because it requires the collection of accurate personal data of citizens. Privacy concerns and the lack of data of crime events at the individual level also need to be taken into account. This problem can be solved by the application of agent-based modeling and simulation. The goal of this work was to create a computer simulation that models human routine activities and facilitates the occurrence of criminal acts as interactions of agents in order to gain a better understanding of the underlying phenomena. The implemented computer simulation was run in several controlled experiments to generate data at the individual and at the aggregate level. These results contributed to answering the following two research questions:

> *RQ 1: Is the routine activity theory, which assumes that crime rates decrease when there is more time spent at safe places, valid for an agent-based street crime model representing Maputo City?*

> *RQ 2: Are law enforcement officers, who patrol along the crime hot spots of the standard model, significantly reducing the crime rates in the advanced model? To what extent does relocation of crime to new clusters occur?*

The main research methods are modeling, simulation and evaluation. The study design followed an established scientific methodology, which was adapted from the available literature. First, the research questions were formulated and the theoretical background was researched. During the modeling process, a simplified representation of the reality was deducted. The practical task of this work comprised of the implementation of a computer program based on the specifications provided by the theoretical model. In the next step of the selected approach, the model was calibrated with data from empiric research and official sources. The verification and validation steps were performed iteratively throughout the implementation process. In a sensitivity analysis, the robustness of the model was shown.

In the theoretical part of this thesis, the foundations concerning geographic information systems, agent-based modeling and simulation and the routine activity theory were elaborated and discussed. The research on GIS provided a foundation for integrating real map data of Maputo City into a computer simulation and to realize the routine activity spaces of citizens. The conducted research on computer simulation presented an overview of simulation approaches and methods, a discussion about the scientific applications of computer simulation and a framework for establishing trust in a computer simulation. The assessment of the routine activity theory by Cohen and Felson provided the conceptual basis for developing the Street Crime Model in the next step.

The subject of study, Maputo City, was selected due to a research cooperation of TU Wien and Eduaro Mondlane University in Maputo. The selection of a capital city of a developing country for evaluating the routine activity theory is a new approach, which led to new results applicable to similar cities. The street map of Maputo city served as the simulation environment for the developed Street Crime Model. The simulation was customized to the city of Maputo by use of real map data, official crime rates and census data such as population, unemployment rate, income and wealth.

The resulting model includes human decision-making and behavior such as performing daily routine activities and human interaction in the form of street robbery. Other types of crime are not modeled. Street robbery is classified as an instrumental crime. This type of criminal acts strives for economic gain and therefore are a result of rational choice rather than an expressive crime. Agent rules were defined to approximate rational human decision-making. The main constructs are citizen agents, which can act as offender, victim or guardian. Police agents have the ability to prevent crimes. Agent movement was modeled as circular paths which connect the four activity spaces home, work, shopping and a recreational place.

Building the computer simulation was one of the main goals of this work. The computer program simulates interactions of citizens in a spatio-temporal environment representing Maputo, the capital of Mozambique. One challenge of this task was the integration of real street map data into the agent-based simulation. The software packages Agent Analyst and ArcGIS enabled integration of GIS and ABMS. Other challenges were the realization of agent movement and agent decision-making. Citizen agents moved along predefined routes, which were computed by the network route solver of ArcGIS and stored in CSV files. Agent decision-making followed the classical rational choice approach and comprised of simple rules. An advanced version of the simulation model introduced a new policing strategy where police officers moved along a route which connects hot spots of crime.

Computer simulation is a research method that produces data as a controlled experiment. In this work, results were generated during a total of fourteen simulation runs and over 100 hours of simulation run-time. Results were recorded and saved for citizens, police officers, places and robbery events. This generated data was used for the evaluation of the proposed hypotheses.

In order to evaluate research question one, twelve scenarios were simulated, each with a systematically changed percentage of the day spent in safe places (`TimeSafe`) input parameter, while other parameters remained unchanged. The influence of the `TimeSafe` parameter on the number of committed offenses was investigated by a linear regression model. The regression showed that the influence of the independent variable `TimeSafe` on the dependent variable `CrimeRate` is significant and that the first research question can be answered positively. The main hypothesis of the routine activity theory is valid for the presented agent-based street crime model of Maputo City.

The second research question was examined based on place agent data and by a comparison of crime rates between the two models. The introduced policing strategy of patrolling along the crime hot spots of the standard model proved to be effective and significantly reduced crime rates in the advanced model. The prevented offenses by police officers increased by 608 % and the number of offenses decreased by 16 %. Another finding was that the introduced policing strategy could only eliminate crime to a certain extent. This observed crime displacement accounted for 45 % of prevented offenses.

The practical relevance of the results exists for several aspects. The propositions of the routine activity are evaluated to be valid for the object of study, Maputo. As a result, the applications and implications of the routine activity theory (see Chapter 2.3.2) apply to Maputo and similar cities. The main consequence of the results would be to avoid the convergence of the three elements of crime and to minimize the time at risk of being victimized. If only one of the three elements of crime is missing, no crime takes place. Crime prevention strategies can target each of the three elements. This work provides a scientific foundation for city administrations and law enforcement organizations to evaluate and refine their crime prevention policies. The recommendation for individuals is to avoid being a suitable victim. Individuals could also seek the presence of guardians and stay in groups or in busy streets. Possible measures of city administrations are improvements of street lighting, CCTV surveillance and situational crime prevention, which aims at preventing the three elements of crime to converge. This work provides a tool for law enforcement organizations to evaluate different strategies of hot spot policing and to predict relocation of crime.

8.2 Future Work and Next Steps

The development of a simulation model is an iterative process and the model improves with each iteration and with each completed verification and validation task. The modeling process involves constant decision-making about whether to include an aspect of the reality and at which granularity. As a result, the simulation model is always a compromise of required complexity to imitate the reality and simplicity to fulfill constraints regarding time and computing resources. A more complex model would produce results with a higher credibility. So one approach for future work could be a more sophisticated model. The presented limitations of the work in the last chapter (see Section 7.4) would be a starting point for improvements. The introduction of the time of day for citizens would bring more realism to their everyday routines. The model would benefit from more detailed GIS and census data, for example land use and population density. Complex agents as described in Section 2.2.5 would lead to a more realistic behavior. For example the use of genetic algorithms to account for past experience would be feasible. Citizens would probably avoid certain streets at night, like in the real world.

The advanced model of this work introduces one specific type of hot spot policing. Possible future work could implement several types of policing strategies and investigate the effectiveness and side effects of different policing strategies. The spatial patterns of crime would be another field of research. The impact of districts with different wealth and income distributions on crime patterns could be investigated.

There are other potential applications of the presented solution. The work can be adapted for research of criminality in other cities, or even regions. Changing the underlying street network requires some preparations including the generation of the predefined activity spaces, but the simulation program works with any street network as input. From a generic point of view, the simulation imitates agent interaction in a network. The criminal act of robbery as a research topic of this thesis is a special type of agent interaction. Other types of agent interaction could be implemented with minor changes. An agent-based model of a virus pandemic is a possible variation and a relevant research area. For example a *spread disease*-method replaces the *rob*-method in order to evaluate different strategies of COVID-19 counter-measures.

The presented work offers a means to produce empiric data by computer simulation for evaluating theories in the area of social science. Many other topics of different disciplines could be studied by using this approach.

List of Figures

List of Tables

Bibliography

[AH81] Robert Axelrod and William Donald Hamilton. „The evolution of cooperation“. In: *science* 211.4489 (1981), pp. 1390–1396.

[AHNV13] Jamal Jokar Arsanjani, Marco Helbich, and Eric de Noronha Vaz. „Spatiotemporal simulation of urban growth patterns using agent-based modeling: The case of Tehran“. In: *Cities* 32 (2013), pp. 33–42.

[AR15] Benjamin Auer and Horst Rottmann. *Statistik und Ökonometrie für Wirtschaftswissenschaftler*. Wiesbaden: Springer Fachmedien Wiesbaden, 2015. ISBN: 978-3-658-06438-9 978-3-658-06439-6. URL: http://link.springer.com/10.1007/978-3-658-06439-6.

[AS12] Ronald L. Akers and Christine S. Sellers. *Criminological Theories: Introduction, Evaluation, and Application*. 6th edition. New York: Oxford University Press, Oct. 26, 2012. 432 pp. ISBN: 978-0-19-984448-7.

[Aba+17] Sameera Abar et al. „Agent Based Modelling and Simulation tools: A review of the state-of-art software“. In: *Computer Science Review* 24 (May 1, 2017), pp. 13–33. ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2017.03.001. URL: https://www.sciencedirect.com/science/article/pii/S1574013716301198 (visited on 03/18/2021).

[All17] Allin Cottrell. *gretl*. Gnu Regression, Econometrics and Time-series Library. 2017. URL: http://gretl.sourceforge.net/ (visited on 08/21/2020).

[Ame98] American Institute of Aeronautics and Astronautics. „AIAA guide for the verification and validation of computational fluid dynamics simulations“. In: (1998). DOI: https://arc.aiaa.org/doi/book/10.2514/4.472855. (Visited on 03/18/2021).

[Amr14] S. Amrutha. „Agent based simulation of routine activity with social learning behavior“. In: *2014 IEEE International Conference on Computational Intelligence and Computing Research*. 2014 IEEE International Conference on Computational Intelligence and Computing Research. Dec. 2014, pp. 1–6. DOI: 10.1109/ICCIC.2014.7238320.

[And+03] Christophe Andrieu et al. „An introduction to MCMC for machine learning“. In: *Machine learning* 50.1 (2003), pp. 5–43.

[Ant+92] Marc Antonini et al. „Image coding using wavelet transform". In: *IEEE Transactions on image processing* 1.2 (1992), pp. 205–220.

[Arg18] Argonne National Laboratory. *Repast.* The Repast Suite. 2018. URL: https://repast.github.io/ (visited on 03/16/2020).

[Ato14] Atomic Heritage Foundation. *Computing and the Manhattan Project.* Atomic Heritage Foundation. 2014. URL: https://www.atomicheritage.org/history/computing-and-manhattan-project (visited on 12/05/2020).

[Aus15] Austrian Regulatory Authority for Broadcasting and Telecommunications. *Communications Report 2014.* Vienna, 2015. URL: https://www.rtr.at/TKP/aktuelles/publikationen/publikationen/C-Report_2014.pdf (visited on 11/02/2020).

[Aut17] Autodesk Inc. *AutoCAD Map 3D | 3D GIS Mapping Software | Autodesk.* AutoCAD Map 3D. 2017. URL: https://www.autodesk.com/products/autocad-map-3d/overview (visited on 10/29/2020).

[Axe97] Robert Axelrod. „Advancing the art of simulation in the social sciences". In: *Simulating social phenomena.* Springer, 1997, pp. 21–40.

[Axt00] Robert Axtell. „Why agents?: on the varied motivations for agent computing in the social sciences". In: *Center on Social and Economic Dynamics Washington, DC* (2000).

[BB04] Patricia Brantingham and Paul Brantingham. „Computer Simulation as a Tool for Environmental Criminologists". In: *Security Journal* 17.1 (Jan. 1, 2004), pp. 21–30. ISSN: 1743-4645. DOI: 10.1057/palgrave.sj.8340159. URL: https://doi.org/10.1057/palgrave.sj.8340159 (visited on 03/18/2021).

[BB95] Patricia Brantingham and Paul Brantingham. „Criminality of place". In: *European Journal on Criminal Policy and Research* 3.3 (Sept. 1, 1995), pp. 5–26. ISSN: 0928-1371, 1572-9869. DOI: 10.1007/BF02242925. URL: http://link.springer.com/article/10.1007/BF02242925 (visited on 10/27/2020).

[BCRL07] Anthony Bigbee, Claudio Cioffi-Revilla, and Sean Luke. „Replication of Sugarscape using MASON". In: *Agent-Based Approaches in Economic and Social Complex Systems IV.* Springer, 2007, pp. 183–190.

[BG08] Tibor Bosse and Charlotte Gerritsen. „Agent-based simulation of the spatial dynamics of crime: on the interplay between criminal hot spots and reputation". In: Proceedings of the 7th international joint conference on autonomous agents and multiagent systems - Volume 2. International Foundation for Autonomous Agents and Multiagent Systems, Dec. 5, 2008, pp. 1129–1136. ISBN: 978-0-9817381-1-6. URL: http://dl.acm.org/citation.cfm?id=1402298.1402378 (visited on 10/28/2020).

[BG09a] Tibor Bosse and Charlotte Gerritsen. „Comparing Crime Prevention Strategies by Agent-Based Simulation“. In: IEEE, 2009, pp. 491–496. ISBN: 978-0-7695-3801-3. DOI: 10.1109/WI-IAT.2009.200. URL: https://ieeexplore.ieee.org/abstract/document/5285130 (visited on 03/19/2021).

[BG09b] Tibor Bosse and Charlotte Gerritsen. „Social Simulation and Analysis of the Dynamics of Criminal Hot Spots“. In: *Journal of Artificial Societies and Social Simulation* 13.2 (2009), p. 5. ISSN: 1460-7425.

[BJ03] Kate J. Bowers and Shane D. Johnson. „Measuring the Geographical Displacement and Diffusion of Benefit Effects of Crime Prevention Activity“. In: *Journal of Quantitative Criminology* 19.3 (Sept. 1, 2003), pp. 275–301. ISSN: 1573-7799. DOI: 10.1023/A:1024909009240. URL: https://doi.org/10.1023/A:1024909009240 (visited on 04/05/2021).

[BPH12] Anthony Braga, Andrew Papachristos, and David Hureau. „Hot spots policing effects on crime“. In: *Campbell Systematic Reviews* 8.1 (2012), pp. 1–96. ISSN: 1891-1803. DOI: https://doi.org/10.4073/csr.2012.8. URL: https://onlinelibrary.wiley.com/doi/abs/10.4073/csr.2012.8 (visited on 01/26/2021).

[BTS12] Daniel Birks, Michael Townsley, and Anna Stewart. „Generative Explanations of Crime: Using Simulation to Test Criminological Theory*“. In: *Criminology* 50.1 (2012), pp. 221–254. ISSN: 1745-9125. DOI: https://doi.org/10.1111/j.1745-9125.2011.00258.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1745-9125.2011.00258.x (visited on 03/18/2021).

[Bal03] Osman Balci. „Verification, validation, and certification of modeling and simulation applications“. In: *Proceedings of the 35th conference on Winter simulation: driving innovation*. Winter Simulation Conference, 2003, pp. 150–158.

[Bal14] Ball Aerospace & Technologies Corp. *Welcome To Opticks*. 2014. URL: https://web.archive.org/web/20201102220307/https://opticks.org/ (visited on 06/29/2021).

[Bal94] Osman Balci. „Validation, verification, and testing techniques throughout the life cycle of a simulation study“. In: *Simulation Conference Proceedings, 1994. Winter*. Simulation Conference Proceedings, 1994. Winter. Dec. 1994, pp. 215–220. DOI: 10.1109/WSC.1994.717129.

[Bar95] Norbert Bartelme. *Geoinformatik Modelle • Strukturen • Funktionen 4. Auflage*. Berlin: Springer, 1995. ISBN: 978-3-540-20254-7.

[Bed11] Mark A Bedau. „Weak emergence and computer simulation“. In: *Models, simulations, and representations* (2011), pp. 91–114.

[Ben17] Bentley Systems, Inc. *Map - Engineering GIS and Geospatial Mapping Software*. Bentley Map. 2017. URL: https://www.bentley.com/en/products/brands/map (visited on 10/29/2020).

[Ber83] Jacques Bertin. „Semiology of graphics: diagrams, networks, maps“. In: *JSTOR* (1983).

[Bon02] Eric Bonabeau. „Agent-based modeling: Methods and techniques for simulating human systems“. In: *PNAS* 99 (2002), p. 3. URL: https://www.pnas.org/content/99/suppl_3/7280 (visited on 03/19/2021).

[Bor+17] Cenab Batu Bora et al. „Modeling and simulation of the resistance of bacteria to antibiotics“. In: *Periodicals of Engineering and Natural Sciences (PEN)* 5.3 (Oct. 18, 2017). Number: 3. ISSN: 2303-4521. DOI: 10.21533/pen.v5i3.146. URL: http://pen.ius.edu.ba/index.php/pen/article/view/146 (visited on 03/18/2021).

[Bra+05] P. L. Brantingham et al. „A computational model for simulating spatial aspects of crime in urban environments“. In: 2005 IEEE international conference on systems, man and cybernetics. Vol. 4. IEEE, 2005, pp. 3667–3674.

[Bra+19] Anthony A. Braga et al. „Hot spots policing and crime reduction: an update of an ongoing systematic review and meta-analysis“. In: *Journal of Experimental Criminology* 15.3 (Sept. 1, 2019), pp. 289–311. ISSN: 1572-8315. DOI: 10.1007/s11292-019-09372-3. URL: https://doi.org/10.1007/s11292-019-09372-3 (visited on 03/18/2021).

[Bri16] British Geological Survey (BGS). *Geological Survey of England and Wales 1:63,360*. OpenGeoscience | Maps. 2016. URL: http://www.bgs.ac.uk/data/maps/maps.cfc?method=viewRecord&mapId=9894 (visited on 12/01/2020).

[Bri+17] Eli S. Bridge et al. „Using Agent-Based Models to Scale from Individuals to Populations“. In: *Aeroecology*. Ed. by Phillip B. Chilson et al. Cham: Springer International Publishing, 2017, pp. 259–275. ISBN: 978-3-319-68576-2. DOI: 10.1007/978-3-319-68576-2_11. URL: https://doi.org/10.1007/978-3-319-68576-2_11 (visited on 03/18/2021).

[Bun09] Bungartz et al. *Modellbildung und Simulation*. eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. ISBN: 978-3-540-79809-5 978-3-540-79810-1. URL: http://link.springer.com/10.1007/978-3-540-79810-1 (visited on 08/26/2020).

[CC12] Andrew T. Crooks and Christian J. E. Castle. „The Integration of Agent-Based Modelling and Geographical Information for Geospatial Simulation“. In: *Agent-Based Models of Geographical Systems*. Ed. by Alison J. Heppenstall et al. Dordrecht: Springer Netherlands, 2012, pp. 219–251. ISBN: 978-90-481-8926-7 978-90-481-8927-4. URL: https://citeseerx.ist.

psu.edu/viewdoc/download?doi=10.1.1.474.3040&rep=rep1&type=pdf (visited on 10/06/2020).

[CE05] Ronald VG Clarke and John E Eck. *Crime analysis for problem solvers in 60 small steps*. US Department of Justice, Office of Community Oriented Policing Services, 2005.

[CF14] Paul R Cohen and Edward A Feigenbaum. *The handbook of artificial intelligence*. Vol. 3. Butterworth-Heinemann, 2014.

[CF79] Lawrence E. Cohen and Marcus Felson. „Social Change and Crime Rate Trends: A Routine Activity Approach“. In: *American Sociological Review* 44.4 (Aug. 1979), p. 588. ISSN: 00031224. DOI: 10.2307/2094589. URL: http://www.jstor.org/discover/10.2307/2094589?uid=3737528&uid=2&uid=4&sid=21105046693993 (visited on 10/27/2020).

[CN13] Nicholson Collier and Michael North. „Parallel agent-based simulation with repast for high performance computing“. In: *Simulation* 89.10 (2013), pp. 1215–1235. URL: https://journals.sagepub.com/doi/10.1177/0037549712462620.

[CR91] J Terry Coppock and David W Rhind. „The history of GIS“. In: *Geographical information systems: Principles and applications* 1.1 (1991), pp. 21–43.

[Cai12] Caitlin Dempsey. *What is GIS?* GIS Lounge. Mar. 1, 2012. URL: https://www.gislounge.com/what-is-gis/ (visited on 11/30/2020).

[Car14] Kathleen M. Carley. „ORA: A Toolkit for Dynamic Network Analysis and Visualization“. In: *Encyclopedia of Social Network Analysis and Mining*. Ed. by Reda Alhajj and Jon Rokne. New York, NY: Springer New York, 2014, pp. 1219–1228. ISBN: 978-1-4614-6170-8. DOI: 10.1007/978-1-4614-6170-8_309. (Visited on 09/18/2020).

[Cen20] Central Intelligence Agency. *The World Factbook*. 2020. URL: https://www.cia.gov/the-world-factbook/ (visited on 03/19/2021).

[Che15] Chethan S. *Philadelphia in 3D using ArcGIS*. 2015. URL: http://www.staygeo.com/2015/03/arcgis-pro-on-nvidia-daas.html (visited on 08/10/2020).

[Chr06] Nicholas Chrisman. *Charting the unknown: How computer mapping at Harvard became GIS*. Esri Press, 2006.

[Chr11] Chris Zeng. *Sugarscape: Agent-Based Modeling*. Wolfram Demonstrations Project. 2011. URL: http://demonstrations.wolfram.com/SugarscapeAgentBasedModeling/ (visited on 03/19/2021).

[Cia18] Giovanni Luca Ciampaglia. „Fighting fake news: a role for computational social science in the fight against digital misinformation“. In: *Journal of Computational Social Science* 1.1 (Jan. 1, 2018), pp. 147–153. ISSN: 2432-2725. DOI: 10.1007/s42001-017-0005-6. URL: https://doi.org/10.1007/s42001-017-0005-6 (visited on 03/18/2021).

[Com11] Wikimedia Commons. *Illustration of geographic latitude and longitude of the earth.* 2011. URL: https://upload.wikimedia.org/wikipedia/commons/6/62/Latitude_and_Longitude_of_the_Earth.svg (visited on 03/19/2021).

[Cri19] Criminal Justice Information Services Division, Federal Bureau of Investigation. *About Crime in the U.S. (CIUS).* FBI. 2019. URL: https://ucr.fbi.gov/crime-in-the-u.s/2019/crime-in-the-u.s.-2019/topic-pages/robbery (visited on 01/27/2021).

[Cro12] Andrew Crooks. „The Use of Agent-Based Modelling for Studying the Social and Physical Environment of Cities“. In: (2012). URL: https://www.researchgate.net/publication/255622944_The_Use_of_Agent-Based_Modelling_for_Studying_the_Social_and_Physical_Environment_of_Cities (visited on 10/06/2020).

[DCL95] Alexis Drogoul, Bruno Corbara, and Steffen Lal. „MANTA: New Experimental Results on the Emergence of (Artificial) Ant Societies“. In: *Ant Societies." Pp. 190-211 in Artificial Societies: The Computer Simulation of Social Life.* UCL Press, 1995, pp. 190–211.

[DEB07] Jason P Davis, Kathleen M Eisenhardt, and Christopher B Bingham. „Developing theory through simulation methods“. In: *Academy of Management Review* 32.2 (2007). Publisher: Academy of Management Briarcliff Manor, NY 10510, pp. 480–499.

[DG97] Marco Dorigo and Luca Maria Gambardella. „Ant colony system: a cooperative learning approach to the traveling salesman problem“. In: *IEEE Transactions on evolutionary computation* 1.1 (1997), pp. 53–66.

[DW13] Nelson Devia and Richard Weber. „Generating crime data using agent-based simulation“. In: *Computers, Environment and Urban Systems* 42 (2013), pp. 26–41. ISSN: 0198-9715. DOI: 10.1016/j.compenvurbsys.2013.09.001.

[Dig+20] Frank Dignum et al. „Analysing the Combined Health, Social and Economic Impacts of the Coronavirus Pandemic Using Agent-Based Social Simulation“. In: *Minds and Machines* 30.2 (June 1, 2020), pp. 177–194. ISSN: 1572-8641. DOI: 10.1007/s11023-020-09527-6. URL: https://doi.org/10.1007/s11023-020-09527-6 (visited on 03/18/2021).

[Dis17] Disy Informationssysteme GmbH. *Cadenza | Disy Informationssysteme GmbH.* Cadenza. 2017. URL: https://www.disy.net/en/home.html (visited on 10/29/2020).

[Dor+18] Jim Doran et al. „The EOS project: modelling Upper Palaeolithic social change“. In: *Simulating societies.* Routledge, 2018, pp. 195–221.

[Dor97] Jim Doran. „From computer simulation to artificial societies“. In: *Transactions of the Society for Computer Simulation International* 14.2 (1997), pp. 69–77.

[EA96] Joshua M Epstein and Robert Axtell. *Growing artificial societies: social science from the bottom up*. Brookings Institution Press, 1996.

[EKD11] Hans-Friedrich Eckey, Reinhold Kosfeld, and Christian Dreger. *Ökonometrie: Grundlagen - Methoden - Beispiele*. 4., durchges. Aufl. Lehrbuch. OCLC: 756271077. Wiesbaden: Gabler, 2011. 423 pp. ISBN: 978-3-8349-3352-2.

[ES87] John E Eck and William Spelman. „Problem-solving: Problem-oriented policing in Newport News". In: (1987). Publisher: Citeseer.

[ESR15] ESRI, Inc. *Independent Report Highlights Esri as Leader in Global GIS Market*. Newsroom. 2015. URL: `http://www.esri.com/esri-news/releases/15-1qtr/independent-report-highlights-esri-as-leader-in-global-gis-market` (visited on 10/31/2020).

[ESR17a] ESRI, Inc. *About ArcGIS*. About ArcGIS. 2017. URL: `http://www.esri.com/arcgis/about-arcgis` (visited on 10/29/2020).

[ESR17b] ESRI, Inc. *ArcGIS Desktop 10.5: ArcMap Functionality Matrix*. 2017. URL: `https://www.esri.com/~/media/Files/Pdfs/library/whitepapers/pdfs/arcmap-functionality-matrix.pdf` (visited on 11/08/2020).

[Eck+05] J. Eck et al. *Mapping crime: Understanding Hotspots*. Report. Washington DC: National Institute of Justice, Aug. 1, 2005, pp. 1–71. URL: `http://discovery.ucl.ac.uk/11291/` (visited on 08/18/2020).

[Ele14] Electronic Arts Inc. *SimCity*. 2014. URL: `http://www.simcity.com/en_US/game/info/what-is-simcity` (visited on 11/27/2020).

[Els86] Jon Elster. *Rational choice*. NYU Press, 1986.

[Env16a] Environmental Systems Research Institute (Esri). *The 50th Anniversary of GIS | ArcNews*. Esri.com. 2016. URL: `http://www.esri.com/news/arcnews/fall12articles/the-fiftieth-anniversary-of-gis.html` (visited on 12/06/2020).

[Env16b] Environmental Systems Research Institute (Esri). *What is GIS*. Esri.com. 2016. URL: `http://www.esri.com/what-is-gis` (visited on 11/30/2020).

[Eps02] Joshua M. Epstein. „Modeling civil violence: An agent-based computational approach". In: *Proceedings of the National Academy of Sciences of the United States of America* 99 (Suppl 3 2002), pp. 7243–7250. URL: `https://www.pnas.org/content/99/suppl_3/7243` (visited on 03/19/2021).

[Eps06] Joshua M. Epstein. *Generative social science: studies in agent-based computational modeling*. Princeton: Princeton University Press, 2006. ISBN: 0-691-12547-3 978-0-691-12547-3.

[Esr98] Esri, Inc. *ESRI Shapefile Technical Description*. 1998. URL: https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/shapefile.pdf (visited on 03/19/2021).

[FDP08] Jacques Ferber, Jean-Louis Dessalles, and Denis Phan. „Emergence in Agent based Computational Social Science: conceptual, formal and diagrammatic analysis“. In: *Intelligent complex adaptative systems* (2008). URL: http://hal.archives-ouvertes.fr/lirmm-00344351/ (visited on 10/06/2020).

[Fon+10] Maria Fonoberova et al. „Nonlinear Dynamics of Crime and Violence in Urban Settings“. In: *Journal of Artificial Societies and Social Simulation* 15.1 (2010), p. 2. ISSN: 1460-7425.

[For71] Jay W Forrester. *World dynamics*. Wright-Allen Press, 1971.

[GJT19] Elizabeth R. Groff, Shane D. Johnson, and Amy Thornton. „State of the Art in Agent-Based Modeling of Urban Crime: An Overview“. In: *Journal of Quantitative Criminology* 35.1 (Mar. 1, 2019), pp. 155–193. ISSN: 1573-7799. DOI: 10.1007/s10940-018-9376-y. URL: https://doi.org/10.1007/s10940-018-9376-y (visited on 03/18/2021).

[GM08] Elizabeth Groff and Lorraine Mazerolle. „Simulated experiments and their potential role in criminology and criminal justice“. In: *Journal of Experimental Criminology* 4.3 (Aug. 27, 2008), p. 187. ISSN: 1572-8315. DOI: 10.1007/s11292-008-9058-0. URL: https://doi.org/10.1007/s11292-008-9058-0 (visited on 09/14/2020).

[GRA15] GRASS Development Team. *GRASS GIS - General overview*. GRASS GIS. 2015. URL: https://grass.osgeo.org/documentation/general-overview/ (visited on 10/28/2020).

[GT00] Nigel Gilbert and Pietro Terna. „How to build and use agent-based models in social science“. In: *Mind & Society* 1.1 (2000), pp. 57–72. URL: http://link.springer.com/article/10.1007/BF02512229 (visited on 08/26/2020).

[GT05] Nigel Gilbert and Klaus G Troitzsch. *Simulation for the Social Scientist*. Open University Press, 2005. ISBN: 0-335-21600-5.

[Gar08] David Garland. „On the concept of moral panic“. In: *Crime, Media, Culture* 4.1 (2008). Publisher: SAGE Publications Sage UK: London, England, pp. 9–30.

[Gar70] Martin Gardner. „Mathematical games: The fantastic combinations of John Conway's new solitaire game “life”“. In: *Scientific American* 223.4 (1970), pp. 120–123.

[Gil02] G. Nigel Gilbert. „Varieties of emergence“. In: *Agent 2002 Conference: Social agents: ecology, exchange, and evolution, Chicago*. 2002, pp. 11–12. URL: https://ccl.northwestern.edu/2002/Gilbert_ABM_Agent2002.pdf (visited on 08/26/2020).

[Gil08] G. Nigel Gilbert. *Agent-based models*. Los Angeles: Sage Publications, 2008. ISBN: 978-1-4129-4964-4 1-4129-4964-5.

[Gil10] G. Nigel Gilbert, ed. *Computational Social Science*. Sage benchmarks in social research methods. OCLC: ocn499090168. Los Angeles: SAGE, 2010. 4 pp. ISBN: 978-1-84787-171-8.

[Gil96] G Nigel Gilbert. „Simulation as a research strategy“. In: *Social science microsimulation*. Springer, 1996, pp. 448–454.

[Goo92] Michael F Goodchild. „Geographical information science“. In: *International journal of geographical information systems* 6.1 (1992), pp. 31–45.

[Gou69] Leroy C Gould. „The changing structure of property crime in an affluent society“. In: *Social Forces* 48.1 (1969). Publisher: The University of North Carolina Press, pp. 50–59.

[Gov08] Government of Canada. *Spatial Resolution, Pixel Size, and Scale*. Natural Resources Canada. Jan. 29, 2008. URL: http://www.nrcan.gc.ca/node/9407 (visited on 08/16/2020).

[Gre13] Thomas Grechenig. *Report on requirements and impact analysis*. B.1. Vienna: Vienna University of Technology, 2013. URL: https://www.inso.tuwien.ac.at/ (visited on 03/19/2021).

[Gro06] Elizabeth Groff. „Exploring the Geography of Routine Activity Theory: A Spatio-Temporal Test Using Street Robbery“. In: (July 10, 2006). URL: http://drum.lib.umd.edu/handle/1903/3775 (visited on 12/17/2020).

[Gro07a] Elizabeth R Groff. „Simulation for theory testing and experimentation: An example using routine activity theory and street robbery“. In: *Journal of Quantitative Criminology* 23.2 (2007). Publisher: Springer, pp. 75–103. DOI: https://doi.org/10.1007/s10940-006-9021-z.

[Gro07b] Elizabeth R Groff. „’Situating’ Simulation to Model Human Spatio-Temporal Interactions: An Example Using Crime Events“. In: *Transactions in GIS* 11.4 (2007), pp. 507–530. ISSN: 1467-9671. DOI: 10.1111/j.1467-9671.2007.01058.x. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9671.2007.01058.x/abstract (visited on 12/17/2020).

[Gro08] Elizabeth R Groff. „Adding the temporal and spatial aspects of routine activities: A further test of routine activity theory“. In: *Security Journal* 21.1 (2008). Publisher: Springer, pp. 95–116.

[Gui] *Eclipse Project | The Eclipse Foundation*. In collab. with Christopher Guindon. 2019. URL: https://www.eclipse.org/eclipse/ (visited on 09/11/2020).

[HB20] Carol M. Huttar and Karlynn BrintzenhofeSzoc. „Virtual Reality and Computer Simulation in Social Work Education: A Systematic Review“. In: *Journal of Social Work Education* 56.1 (Jan. 2, 2020). Publisher: Routledge _eprint: https://doi.org/10.1080/10437797.2019.1648221, pp. 131–141. ISSN: 1043-7797. DOI: 10.1080/10437797.2019.1648221. URL: https://doi.org/10.1080/10437797.2019.1648221 (visited on 03/18/2021).

[HCS11] Alison J. Heppenstall, Andrew T. Crooks, and Linda M. See. *Agent-Based Models of Geographical Systems*. Springer, Nov. 24, 2011. 747 pp. ISBN: 978-90-481-8927-4.

[HRE03] David Hales, Juliette Rouchier, and Bruce Edmonds. „Model-to-model analysis“. In: *Journal of Artificial Societies and Social Simulation* 6.4 (2003).

[HRWL84] Frederick Hayes-Roth, Donald Waterman, and Douglas Lenat. „Building expert systems“. In: *Teknowledge Series in Knowledge Engineering.* (1984), pp. 405–420.

[HT15] Roland Maximilian Happach and Meike Tilebein. „Simulation as Research Method: Modeling Social Interactions in Management Science“. In: *Collective Agency and Cooperation in Natural and Artificial Systems: Explanation, Implementation and Simulation*. Ed. by Catrin Misselhorn. Philosophical Studies Series. Cham: Springer International Publishing, 2015, pp. 239–259. ISBN: 978-3-319-15515-9. DOI: 10.1007/978-3-319-15515-9_13. URL: https://doi.org/10.1007/978-3-319-15515-9_13 (visited on 03/17/2021).

[Har91] Ann Harding. „Dynamic microsimulation models: problems and prospects“. In: *Centre for Analysis of Social Exclusion, The London School of Economics and Political Science* (1991).

[He+06] Chunyang He et al. „Modeling urban expansion scenarios by coupling cellular automata model and system dynamic model in Beijing, China“. In: *Applied Geography* 26.3 (2006), pp. 323–345.

[Hel16] Miguel Helft. *The Godfather of Digital Maps*. Forbes. 2016. URL: https://www.forbes.com/sites/miguelhelft/2016/02/10/the-godfather-of-digital-maps/ (visited on 11/07/2020).

[Her17] Hertfordshire County Council. *Webmaps - Spatial Data Models*. Hertfordshire County Council. 2017. URL: http://gisinfo.hertfordshire.gov.uk/GISdata/vectorraster.htm (visited on 08/12/2020).

[Hev+04] Alan R Hevner et al. „Design science in information systems research“. In: *MIS quarterly* (2004). Publisher: JSTOR, pp. 75–105.

[Hij11] Robert Hijmans. *DIVA-GIS*. free, simple and effective. 2011. URL: http://www.diva-gis.org/ (visited on 10/29/2020).

[Hil07] Amy Hillier. „ArcGIS 9.3 manual“. In: *University of Pennsylvania* (2007). URL: https://works.bepress.com/amy_hillier/17/ (visited on 03/19/2021).

[Hil09] Linda L Hill. *Georeferencing: The geographic associations of information*. Mit Press, 2009.

[Hom96] Ross Homel. *The politics and practice of situational crime prevention*. Vol. 5. Criminal Justice Press Monsey, NY, 1996.

[Hum90] Paul Humphreys. „Computer simulations“. In: *PSA: Proceedings of the biennial meeting of the philosophy of science association*. Vol. 1990. Philosophy of Science Association, 1990, pp. 497–506.

[ISM17] Hafiz Ali Imran, Dietrich Schröder, and Bilal Ahmed Munir. „Agent-based simulation for biogas power plant potential in Schwarzwald-Baar-Kreis, Germany: a step towards better economy“. In: *Geocarto International* 32.1 (Jan. 2, 2017), pp. 59–70. ISSN: 1010-6049. DOI: 10.1080/10106049.2015.1128485. URL: https://doi.org/10.1080/10106049.2015.1128485 (visited on 03/18/2021).

[Ida08] Idaho State University Department of Geosciences. *Public Land Survey System*. Public Land Survey System. 2008. URL: http://geology.isu.edu/wapi/geostac/Field_Exercise/topomaps/plss.htm (visited on 09/01/2020).

[Ins15] Instituto Nacional de Estatística – Moçambique. *Estatísticas de Crime e Justiça*. 2015. URL: http://www.ine.gov.mz/estatisticas/estatisticas-sectoriais/crime-e-justica/crime-e-justica-2013-2014 (visited on 03/19/2021).

[Int18] International Monetary Fund. *IMF Data*. Report for Selected Countries and Subjects. Library Catalog: www.imf.org. 2018. URL: https://www.imf.org/en/Data (visited on 06/13/2020).

[JVR17] JUMP Pilot Project, Vivid Solutions, and Refractions Research. *OpenJUMP GIS*. OpenJUMP GIS. 2017. URL: http://www.openjump.org/ (visited on 10/29/2020).

[JW96] Dennis E Jelinski and Jianguo Wu. „The modifiable areal unit problem and implications for landscape ecology“. In: *Landscape ecology* 11.3 (1996), pp. 129–140.

[Jam08] James Gray. *Getting Started With Quantum GIS | Linux Journal*. Getting Started With Quantum GIS. 2008. URL: https://www.linuxjournal.com/content/getting-started-quantum-gis (visited on 10/29/2020).

[Jen67] George F Jenks. „The data model concept in statistical mapping“. In: *International yearbook of cartography* 7 (1967), pp. 186–190.

[Joh02] Ron Johnston. „Manipulating maps and winning elections: measuring the impact of malapportionment and gerrymandering“. In: *Political geography* 21.1 (2002), pp. 1–31.

[Joh13] Kevin M. Johnston. *Agent Analyst: Agent-Based Modeling in ArcGIS | ArcGIS Resources*. First Edition. 380 New York Street, Redlands, California: Esri Press, 2013. URL: `http://resources.arcgis.com/en/help/agent-analyst/` (visited on 07/06/2020).

[KJ20] Amos Kalua and James Jones. „Epistemological Framework for Computer Simulations in Building Science Research: Insights from Theory and Practice“. In: *Philosophies* 5.4 (2020). Publisher: Multidisciplinary Digital Publishing Institute, p. 30.

[Klu08] Franziska Kluegl. „A validation methodology for agent-based simulations“. In: *Proceedings of the 2008 ACM symposium on Applied computing*. 2008, pp. 39–43.

[LDJ19] Johanna Leigh, Sarah Dunnett, and Lisa Jackson. „Predictive police patrolling to target hotspots and cover response demand“. In: *Annals of Operations Research* 283.1 (Dec. 1, 2019), pp. 395–410. ISSN: 1572-9338. DOI: `10.1007/s10479-017-2528-x`. URL: `https://doi.org/10.1007/s10479-017-2528-x` (visited on 03/18/2021).

[LMOM12] Federico Lievano Martinez and Yris Olaya Morales. „Agent-based simulation approach to urban dynamics modeling“. In: *Dyna* 79.173 (2012), pp. 34–42.

[LT03] Wolfgang Loibl and Tanja Toetzer. „Modeling growth and densification processes in suburban regions—simulation of landscape transition with spatial agents“. In: *Environmental Modelling & Software* 18.6 (2003), pp. 553–563.

[LY10] Jay Lee and Chaoqing Yu. „The development of urban crime simulator“. In: *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application*. ACM, 2010, p. 60. URL: `http://dl.acm.org/citation.cfm?id=1823921` (visited on 03/19/2021).

[Lar+99] Larranaga et al. „Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators“. In: *Artificial Intelligence Review* 13.2 (1999), p. 42.

[Lem11] Mathias Lemmens. „Terrestrial laser scanning“. In: *Geo-information*. Springer, 2011, pp. 101–121.

[Lon+15] Paul A Longley et al. *Geographic information science and systems*. Fourth Edition. Hoboken, NJ: John Wiley & Sons, 2015. ISBN: 978-1-119-03130-7.

[Lou+18] P. Lou et al. „Behavior Simulation of Manufacturing Services in a Cloud Manufacturing Environment". In: *2018 3rd International Conference on Information Systems Engineering (ICISE)*. 2018 3rd International Conference on Information Systems Engineering (ICISE). ISSN: 2160-1291. May 2018, pp. 137–141. DOI: `10.1109/ICISE.2018.00033`.

[Lyn87] James P. Lynch. „Routine Activity and Victimization at Work". In: *Journal of Quantitative Criminology* 3.4 (1987), pp. 283–300. ISSN: 0748-4518. URL: `www.jstor.org/stable/23365566` (visited on 11/23/2020).

[MB12] Nick Malleson and Mark Birkin. „Analysis of crime patterns through the integration of an agent-based model and a population microsimulation". In: *Computers, Environment and Urban Systems* 36.6 (Nov. 2012), pp. 551–561. ISSN: 01989715. DOI: `10.1016/j.compenvurbsys.2012.04.003`. URL: `http://www.nickmalleson.co.uk/wp-content/uploads/2012/01/geocomp-ceus.pdf` (visited on 03/19/2021).

[ME06] Pratap Misra and Per Enge. *Global Positioning System: Signals, Measurements and Performance*. Second Edition. Lincoln, MA: Ganga-Jamuna Press, 2006.

[MHS10] Nick Malleson, Alison Heppenstall, and Linda See. „Crime reduction through simulation: An agent-based model of burglary". In: *Computers, Environment and Urban Systems* 34.3 (2010), pp. 236–250. DOI: `10.1016/j.compenvurbsys.2009.10.005`.

[MHS95] Hernán A Makse, Shlomo Havlin, and H Eugene Stanley. „Modelling urban growth patterns". In: *Nature* 377.6550 (1995), pp. 608–612.

[MN08] Charles M. Macal and Michael J. North. „Agent-based modeling and simulation: ABMS examples". In: *Proceedings of the 40th Conference on Winter Simulation*. 2008, pp. 101–112. URL: `http://dl.acm.org/citation.cfm?id=1516770` (visited on 10/06/2020).

[MN09] Charles M. Macal and Michael J. North. „Agent-based modeling and simulation". In: IEEE, Dec. 2009, pp. 86–98. ISBN: 978-1-4244-5770-0. DOI: `10.5555/1995456.1995474`. URL: `https://dl.acm.org/doi/10.5555/1995456.1995474` (visited on 03/19/2021).

[MN10] Charles M. Macal and Michael J. North. „Toward teaching agent-based simulation". In: *Simulation Conference (WSC), Proceedings of the 2010 Winter*. 2010, pp. 268–277. URL: `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5679158` (visited on 10/06/2020).

[MSN04] C Macal, D Sallach, and M North. „Emergent structures from trust relationships in supply chains". In: *Proc. Agent 2004: Conf. on Social Dynamics*. 2004, pp. 7–9.

[MT99] Elizabeth Ehrhardt Mustaine and Richard Tewksbury. „A Routine Activity Theory Explanation for Women's Stalking Victimizations". In: *Violence Against Women* 5.1 (Jan. 1, 1999). Publisher: SAGE Publications Inc, pp. 43–62. ISSN: 1077-8012. DOI: 10.1177/10778019922181149. URL: https://doi.org/10.1177/10778019922181149 (visited on 01/26/2021).

[MV80] Humberto R Maturana and Francisco J Varela. „Autopoiesis and cognition: The realization of the living". In: *Autopoiesis and cognition*. Springer, 1980, pp. 73–76.

[MW02] Michael W. Macy and Robert Willer. „From factors to actors: Computational sociology and agent-based modeling". In: *Annual review of sociology* (2002), pp. 143–166. URL: http://www.jstor.org/stable/3069238 (visited on 08/26/2020).

[Mac95] Alan M MacEachren. *How maps work: representation, visualization, and design*. Guilford Press, 1995.

[Mai+13] David Maimon et al. „Daily trends and origin of computer-focused crimes against a large university computer network: An application of the routine-activities and lifestyle perspective". In: *British Journal of Criminology* 53.2 (2013). Publisher: Oxford University Press UK, pp. 319–343.

[Mal+13] Nick Malleson et al. „Using an agent-based crime simulation to predict the effects of urban regeneration on individual household burglary risk". In: *Environment and Planning B: Planning and Design* 40.3 (2013), pp. 405–426. ISSN: 0265-8135, 1472-3417. DOI: 10.1068/b38057. URL: http://www.envplan.com/abstract.cgi?id=b38057 (visited on 10/28/2020).

[Map17] MapWindow Developers Team. *About the MapWindow GIS Open Source Project*. The MapWindow project - Home. 2017. URL: http://www.mapwindow.org/#about (visited on 10/29/2020).

[Mar+00] Michael W Marcellin et al. „An overview of JPEG-2000". In: *Data Compression Conference, 2000. Proceedings. DCC 2000.* IEEE, 2000, pp. 523–541.

[Mel+18] Silas Nogueira de Melo et al. „Spatial/Temporal Variations of Crime: A Routine Activity Theory Perspective". In: *International Journal of Offender Therapy and Comparative Criminology* 62.7 (May 1, 2018). Publisher: SAGE Publications Inc, pp. 1967–1991. ISSN: 0306-624X. DOI: 10.1177/0306624X17703654. URL: https://doi.org/10.1177/0306624X17703654 (visited on 03/22/2021).

[Mic83] Ryszard S Michalski. „A theory and methodology of inductive learning". In: *Artificial intelligence* 20.2 (1983), pp. 111–161.

[Mih17] Regina Mihindukulasuriya. *Esri's Annual Revenues Exceed $1.1 Billion*. BW Businessworld. 2017. URL: http://businessworld.in/article/-Esri-s-Annual-Revenues-Exceed-1-1-Billion-/24-04-2017-116907 (visited on 11/07/2020).

[Mil98] John H. Miller. „Active Nonlinear Tests (ANTs) of Complex Simulation Models“. In: *Management Science* 44.6 (1998), pp. 820–830. DOI: 10.1287/mnsc.44.6.820. URL: https://pubsonline.informs.org/doi/abs/10.1287/mnsc.44.6.820.

[Min+96] Nelson Minar et al. „The swarm simulation system: A toolkit for building multi-agent simulations“. In: *Technical report, Swarm Development Group* (1996).

[Mor50] Patrick Moran. „Notes on continuous stochastic phenomena“. In: *Biometrika* 37.1 (1950), pp. 17–23.

[Mun+20] Bilal Ahmad Munir et al. „Geospatial assessment of physical accessibility of healthcare and agent-based modeling for system efficacy“. In: *GeoJournal* 85.3 (June 1, 2020), pp. 665–680. ISSN: 1572-9893. DOI: 10.1007/s10708-019-09987-z. URL: https://doi.org/10.1007/s10708-019-09987-z (visited on 03/18/2021).

[NAS16] NASA/USGS. *The Landsat Program*. Landsat Science. 2016. URL: http://landsat.gsfc.nasa.gov/ (visited on 12/01/2020).

[NCV06] Michael J North, Nicholson T Collier, and Jerry R Vos. „Experiences creating three implementations of the repast agent modeling toolkit“. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 16.1 (2006), pp. 1–25.

[ND17] N Alghais and D Pullar. „Modelling future impacts of urban development in Kuwait with the use of ABM and GIS“. In: *Transactions in GIS*. Wiley Online Library, 2017. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.12293 (visited on 03/18/2021).

[NM07] Michael North and Charles Macal. *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation*. Oxford, New York: Oxford University Press, Mar. 22, 2007. 313 pp. ISBN: 978-0-19-517211-9.

[NM09] Cynthia Nikolai and Gregory Madey. *Tools of the Trade: A Survey of Various Agent Based Modeling Platforms*. Mar. 31, 2009. URL: http://jasss.soc.surrey.ac.uk/12/2/2.html (visited on 09/12/2020).

[NS76] Allen Newell and Herbert A Simon. *Computer Science as Empirical Inquiry: Symbols and Search*. Vol. 19. Communications of the ACM 3. ACM, 1976. 113–126.

[Nat00] National Imagery and Mapping Agency (NIMA). „Department of defense world geodetic system 1984, its definition and relationships with local geodetic systems“. In: *National Geospatial-Intelligence Agency, Tech. Rep* 152 (2000).

[Nat12] National Oceanic and Atmospheric Administration US Department of Commerce. *What is LIDAR*. 2012. URL: https://oceanservice.noaa.gov/facts/lidar.html (visited on 09/12/2020).

[Net00] Markus Neteler. „GRASS-handbuch“. In: *Der praktische Leitfaden zum Geographischen Informationssystem GRASS* (2000).

[Nor+07] Michael J North et al. *Visual agent-based model development with repast simphony*. Tech. rep., Argonne National Laboratory, 2007.

[Nor+13] Michael J North et al. „Complex adaptive systems modeling with Repast Simphony“. In: *Complex adaptive systems modeling* 1.1 (2013), p. 3. DOI: 10.1186/2194-3206-1-3. URL: https://link.springer.com/article/10.1186/2194-3206-1-3 (visited on 03/18/2021).

[Nor15] John Norton. „An introduction to sensitivity assessment of simulation models“. In: *Environmental Modelling & Software* 69 (2015), pp. 166–174.

[O'L00] John O'Looney. „Beyond Maps“. In: *GIS and Decision Making in Local Government. Redlands* (2000).

[OO81] Stan Openshaw and S Openshaw. „The modifiable areal unit problem“. In: *Quantitative geography: A British view* (1981), pp. 60–69.

[OR10] William L Oberkampf and Christopher J Roy. *Verification and validation in scientific computing*. Cambridge University Press, 2010.

[Ore+94] Naomi Oreskes et al. „Verification, validation, and confirmation of numerical models in the earth sciences“. In: *Science* 263.5147 (1994), pp. 641–646.

[Oxf16] Oxford University Press. *geographic information system - English | Oxford Dictionaries*. In: *Oxford Dictionaries | English*. 2016. URL: https://en.oxforddictionaries.com/definition/geographic_information_system (visited on 11/30/2020).

[Oxf17] Oxford University Press. *simulation | Definition of simulation in English by Oxford Dictionaries*. Oxford Dictionaries | English. 2017. URL: https://en.oxforddictionaries.com/definition/simulation (visited on 11/13/2020).

[Ozi+13] J. Ozik et al. „The ReLogo agent-based modeling language“. In: *2013 Winter Simulations Conference (WSC)*. 2013 Winter Simulations Conference (WSC). Dec. 2013, pp. 1560–1568. DOI: 10.1109/WSC.2013.6721539.

[Ozi+15] Jonathan Ozik et al. „Repast Simphony Statecharts“. In: *Journal of Artificial Societies and Social Simulation* 18.3 (2015), p. 11. ISSN: 1460-7425.

[PC75] Thomas K Peucker and Nicholas Chrisman. „Cartographic data structures“. In: *The American Cartographer* 2.1 (1975), pp. 55–69.

[PEH06] Hazel R. Parry, Andrew J. Evans, and Alison J. Heppenstall. *Millions of agents: Parallel simulations with the Repast agent-based toolkit*. New York: Springer, 2006. URL: https://www.academia.edu/414706/Millions_of_Agents_Parallel_Simulations_With_the_RePast_Agent_Based_Toolkit (visited on 03/19/2021).

[PHR10] Travis C. Pratt, Kristy Holtfreter, and Michael D. Reisig. „Routine Online Activity and Internet Fraud Targeting: Extending the Generality of Routine Activity Theory“. In: *Journal of Research in Crime and Delinquency* 47.3 (Aug. 1, 2010). Publisher: SAGE Publications Inc, pp. 267–296. ISSN: 0022-4278. DOI: 10.1177/0022427810365903. URL: https://doi.org/10.1177/0022427810365903 (visited on 01/26/2021).

[PSR98] H. Van Dyke Parunak, Robert Savit, and Rick L. Riolo. „Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide“. In: *Multi-Agent Systems and Agent-Based Simulation*. Ed. by Jaime Simão Sichman, Rosaria Conte, and Nigel Gilbert. Lecture Notes in Computer Science 1534. Springer Berlin Heidelberg, July 4, 1998, pp. 10–25. ISBN: 978-3-540-65476-6 978-3-540-49246-7. DOI: 10.1007/10692956_2. URL: http://link.springer.com/chapter/10.1007/10692956_2 (visited on 08/26/2020).

[Par08] Wendy S. Parker. „Franklin, Holmes, and the Epistemology of Computer Simulation“. In: *International Studies in the Philosophy of Science* 22.2 (July 1, 2008), pp. 165–183. ISSN: 0269-8595. DOI: 10.1080/02698590802496722. URL: https://doi.org/10.1080/02698590802496722 (visited on 05/23/2020).

[Par09] Emmanuel Paradis. „Moran's autocorrelation coefficient in comparative methods“. In: *R Foundation for Statistical Computing* (2009).

[Pau17] Paul Waddell. *UrbanSim*. UrbanSim. 2017. URL: http://www.urbansim.com/urbansim/ (visited on 11/27/2020).

[Paw01] A Pawlowska. „GIS as a Tool in Local Policy-Making“. In: *Sklodowska University, Lublin* (2001).

[Po06] Scott E Page and others. „Path dependence“. In: *Quarterly Journal of Political Science* 1.1 (2006), pp. 87–115.

[Pre21] Precisely. *MapInfo Pro™ - Desktop GIS*. Precisely. 2021. URL: https://www.precisely.com/product/precisely-mapinfo/mapinfo-pro (visited on 03/19/2021).

[QGI17] QGIS Development Team. *Discover QGIS*. QGIS - The Leading Open Source Desktop GIS. 2017. URL: http://www.qgis.org/en/site/about/index.html (visited on 10/28/2020).

[Qui+05] Sergio E. Quijada et al. „A spatio temporal simulation model for evaluating delinquency and crime policies“. In: Proceedings of the Winter Simulation Conference. IEEE, 2005.

[RG12] Steven F Railsback and Volker Grimm. *Agent-based and individual-based modeling: a practical introduction*. Princeton: Princeton University Press, 2012. ISBN: 978-0-691-13674-5.

[RY17] S. Rui and Z. Yong. „Modelling and Simulation for Rumor Propagation on Complex Networks with Repast Simulation Platform". In: *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. 2017 4th International Conference on Information Science and Control Engineering (ICISCE). July 2017, pp. 1014–1018. DOI: 10.1109/ICISCE.2017.213.

[Rep74] Thomas A Reppetto. *Residential crime*. Ballinger Publishing Company, 1974.

[Rob67] Arthur H Robinson. „Psychological aspects of color in cartography". In: *International Yearbook of Cartography* 7 (1967), pp. 50–61.

[Rob+95] Arthur H Robinson et al. *Elements of Cartography*. 6th. New York: Wiley, 1995. ISBN: 978-0-471-55579-7.

[SBG14] Nils Schuhmacher, Laura Ballato, and Paul van Geert. „Using an agent-based model to simulate the development of risk behaviors during adolescence". In: *Journal of Artificial Societies and Social Simulation* 17.3 (2014), p. 1. URL: http://jasss.soc.surrey.ac.uk/17/3/1.html.

[SLR07] Tilman A. Schenk, Günter Löffler, and Jürgen Rauh. „Agent-based simulation of consumer behavior in grocery shopping on a regional level". In: *Journal of Business Research* 60.8 (Aug. 2007), pp. 894–903. ISSN: 01482963. DOI: 10.1016/j.jbusres.2007.02.005. URL: https://www.sciencedirect.com/science/article/pii/S014829630700046X (visited on 03/19/2021).

[SM06] Douglas A Samuelson and Charles M Macal. „Agent-based simulation comes of age". In: *OR/MS Today* 33.4 (2006).

[STA21] STATISTIK AUSTRIA. *Übersicht der Bundesländer*. Übersicht der Bundesländer. 2021. URL: http://www.statistik.at/web_de/klassifikationen/regionale_gliederungen/bundeslaender/index.html (visited on 03/19/2021).

[Saw03] R. Keith Sawyer. „Artificial Societies Multiagent Systems and the Micro-Macro Link in Sociological Theory". In: *Sociological Methods & Research* 31.3 (Jan. 2, 2003), pp. 325–363. ISSN: 0049-1241, 1552-8294. DOI: 10.1177/0049124102239079. URL: http://smr.sagepub.com/content/31/3/325 (visited on 08/26/2020).

[Sch69] Thomas C Schelling. „Models of segregation". In: *The American Economic Review* 59.2 (1969), pp. 488–493.

[Sch71] Thomas C Schelling. „Dynamic models of segregation". In: *Journal of mathematical sociology* 1.2 (1971), pp. 143–186.

[Sho82] Nicholas M Short. *The LANDSAT Tutorial Workbook: Basics of Satellite Remote Sensing*. Vol. 1078. National Aeronautics, Space Administration, Scientific, and Technical Information Branch, 1982.

[Sia+17] P. Siahaan et al. „Improving Students' Science Process Skills through Simple Computer Simulations on Linear Motion Conceptions". In: *Journal of Physics: Conference Series* 812 (Feb. 2017). Publisher: IOP Publishing, p. 012017. ISSN: 1742-6596. DOI: `10.1088/1742-6596/812/1/012017`. URL: `https://doi.org/10.1088/1742-6596/812/1/012017` (visited on 03/18/2021).

[Sie10] Larry J. Siegel. *Criminology: The Core*. 4th edition. Australia : Belmont, CA: Cengage Learning, Feb. 23, 2010. 512 pp. ISBN: 978-0-495-80983-8.

[Sil+20] Petrônio C. L. Silva et al. „COVID-ABS: An agent-based model of COVID-19 epidemic to simulate health and economic effects of social distancing interventions". In: *Chaos, Solitons & Fractals* 139 (Oct. 1, 2020), p. 110088. ISSN: 0960-0779. DOI: `10.1016/j.chaos.2020.110088`. URL: `https://www.sciencedirect.com/science/article/pii/S0960077920304859` (visited on 03/18/2021).

[Sim97] Herbert Alexander Simon. *Models of bounded rationality: Empirically grounded economic reason*. Vol. 3. MIT press, 1997.

[Squ12] Flaminio Squazzoni. *Agent-based computational sociology*. Hoboken, N.J.: Wiley & Sons, 2012. ISBN: 978-1-119-95419-4. URL: `https://onlinelibrary.wiley.com/doi/book/10.1002/9781119954200` (visited on 03/19/2021).

[Ste00] John Sterman. *Business Dynamics: systems thinking and modeling for a complex world*. 1st. New York, NY, USA: McGraw-Hill, Inc., 2000. ISBN: 0-07-231135-5.

[Ste12] Stephanie Parragh. „Modeling and Simulation with AnyLogic". Institut für Analysis und Scientific Computing. Technische Universität Wien, 2012.

[Str+18] M. Straka et al. „Design of Large-Scale Logistics Systems Using Computer Simulation Hierarchic Structure". In: *International Journal of Simulation Modelling* 17.1 (Mar. 15, 2018), pp. 105–118. ISSN: 17264529. DOI: `10.2507/IJSIMM17(1)422`. URL: `http://www.ijsimm.com/Full_Papers/Fulltext2018/text17-1_105-118.pdf` (visited on 03/18/2021).

[Sui04] Daniel Z Sui. „Tobler's first law of geography: A big idea for a small world?" In: *Annals of the Association of American Geographers* 94.2 (2004), pp. 269–277.

[Sun06] Ron Sun. *Cognition and multi-agent interaction: From cognitive modeling to social simulation*. Cambridge University Press, 2006.

[Sym08] John Symons. „Computational Models of Emergent Properties“. In: *Minds and Machines* 18.4 (Nov. 14, 2008), pp. 475–491. ISSN: 0924-6495, 1572-8641. DOI: 10.1007/s11023-008-9120-8. URL: http://link.springer.com/article/10.1007/s11023-008-9120-8 (visited on 08/26/2020).

[THL15] Lu Tan, Mingyuan Hu, and Hui Lin. „Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency“. In: *Information Sciences* 295 (Feb. 20, 2015), pp. 53–66. ISSN: 0020-0255. DOI: 10.1016/j.ins.2014.09.029. URL: https://www.sciencedirect.com/science/article/pii/S0020025514009359 (visited on 03/18/2021).

[Tei80] Eric Teicholz. „Geographic information systems: the ODYSSEY project“. In: *Journal of the Surveying and Mapping Division* 106.1 (1980), pp. 119–135.

[Teu78] Jukka Teuhola. „A compression method for clustered bit-vectors“. In: *Information processing letters* 7.6 (1978), pp. 308–311.

[The16] The Editors of Encyclopædia Britannica. *information system | Britannica.com*. 2016. URL: https://global.britannica.com/topic/information-system (visited on 11/30/2020).

[The17] The Editors of Encyclopædia Britannica. *computer simulation*. Encyclopedia Britannica. 2017. URL: https://www.britannica.com/technology/computer-simulation (visited on 11/13/2020).

[The19a] The AnyLogic Company. *AnyLogic: Simulation Modeling Software Tools & Solutions for Business*. 2019. URL: https://www.anylogic.com/ (visited on 09/12/2020).

[The19b] The MathWorks Inc. *The MathWorks Home Page*. 2019. URL: https://de.mathworks.com/products/matlab.html (visited on 09/11/2020).

[Tob70] Waldo R Tobler. „A computer movie simulating urban growth in the Detroit region“. In: *Economic geography* 46 (sup1 1970), pp. 234–240.

[Tom69] Roger F Tomlinson. „A Geographic Information System for Regional Planning“. In: *Journal of Geography*. Vol. 78. 1969, pp. 45–48.

[Ton95] Michael H. Tonry, ed. *Building a safer society: strategic approaches to crime prevention*. Crime and Justice 19. Chicago: University of Chicago Press, 1995. viii+704. ISBN: 978-0-226-80824-6.

[US 90] US Bureau of the Census. „Census of Population and Housing“. In: *Public Use Microdata Sample* 5 (1990).

[U.S12] U.S. Department of the Interior. *The Public Land Survey System (PLSS)*. National Atlas of the United States. 2012. URL: https://web.archive.org/web/20130619210714/http://www.nationalatlas.gov/articles/boundaries/a_plss.html (visited on 03/19/2021).

[Uni12] University Consortium for Geographic Information Science. *UCGIS - Mission and Scope*. 2012. URL: https://www.ucgis.org/about-and-mission (visited on 03/19/2021).

[Uni16] United States Geological Survey (USGS). *Geographic Information Systems (GIS) Poster*. 2016. URL: https://web.archive.org/web/20170219004050/https://egsc.usgs.gov/isb//pubs/gis_poster/ (visited on 03/19/2021).

[Uni20] United Nations Statistics Division. *UNdata | country profile | Mozambique*. 2020. URL: http://data.un.org/en/iso/mz.html (visited on 01/04/2021).

[Urb12] Urban Institute. *TRIM3 project website*. 2012. URL: http://trim.urban.org/T3IntroMicrosimulation.php (visited on 12/06/2020).

[VR86] Christy A Visher and Jeffrey A Roth. „Participation in criminal careers“. In: *Criminal careers and career criminals* 1 (1986), pp. 211–291.

[WC14] Sarah Wise and Tao Cheng. „A Model Officer: An Agent-based Model of Policing“. In: *Transactions in GIS* (2014), pp. 680–685. URL: https://discovery.ucl.ac.uk/id/eprint/1492675/1/Cheng_GISRUK2015_submission_48.pdf (visited on 03/19/2021).

[WE93] Roger White and Guy Engelen. „Cellular automata and fractal urban form: a cellular modelling approach to the evolution of urban land-use patterns“. In: *Environment and planning A* 25.8 (1993), pp. 1175–1199.

[Wat17] Nigel Waters. „Tobler's First Law of Geography“. In: *The International Encyclopedia of Geography* (2017).

[Wei+17] David Weisburd et al. „Can hot spots policing reduce crime in urban areas? An agent-based simulation“. In: *Criminology* 55.1 (2017), pp. 137–173.

[Wer08] Josie Wernecke. *The KML handbook: geographic visualization for the Web*. Pearson Education, 2008.

[Wik21] Wikipedia contributors. *List of cities in Africa by population — Wikipedia, The Free Encyclopedia*. 2021. URL: https://en.wikipedia.org/w/index.php?title=List_of_cities_in_Africa_by_population&oldid=1037475788 (visited on 07/01/2021).

[Wil99a] U. Wilensky. *NetLogo Wolf Sheep Predation model*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. 1999. URL: http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation (visited on 03/15/2021).

[Wil99b] U. Wilensky. *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. 1999. URL: https://ccl.northwestern.edu/netlogo/ (visited on 03/15/2021).

[Win03] Eric Winsberg. „Simulated Experiments: Methodology for a Virtual World". In: *Philosophy of Science* 70.1 (Jan. 2003), pp. 105–125. ISSN: 0031-8248, 1539-767X. DOI: 10.1086/367872. URL: http://www.journals.uchicago.edu/doi/10.1086/367872 (visited on 11/14/2020).

[Win07] Peter Winker. *Empirische Wirtschaftsforschung und Ökonometrie: mit 13 Tabellen*. 2., vollst. überarb. Aufl. Springer-Lehrbuch. OCLC: 180942214. Berlin: Springer, 2007. 334 pp. ISBN: 978-3-540-36778-9.

[Win19] Eric Winsberg. „Computer Simulations in Science". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2019. Metaphysics Research Lab, Stanford University, 2019. URL: https://plato.stanford.edu/archives/win2019/entries/simulations-science/.

[Win98] Stephan Winter. „Bridging vector and raster representation in GIS". In: *Proceedings of the 6th ACM international symposium on Advances in geographic information systems*. ACM, 1998, pp. 57–62.

[Wis13] Stephen Wise. *GIS Fundamentals, Second Edition*. CRC Press, Sept. 25, 2013. 340 pp. ISBN: 978-1-4398-8695-3.

[Wol19] Wolfram Research Inc. *Wolfram Mathematica: Moderne technische Berechnung*. 2019. URL: https://www.wolfram.com/mathematica/ (visited on 09/11/2020).

[Won09] David Wong. „The modifiable areal unit problem (MAUP)". In: *The SAGE handbook of spatial analysis* (2009), pp. 105–123.

[ZB13] Yue Zhang and Donald E Brown. „Police patrol districting method and simulation evaluation using agent-based model & GIS". In: *Security Informatics* 2.1 (2013), p. 7. ISSN: 2190-8532. DOI: 10.1186/2190-8532-2-7. URL: http://security-informatics.springeropen.com/articles/10.1186/2190-8532-2-7 (visited on 08/22/2020).

[gvS16] gvSIG Association. *gvSIG Desktop - Portal gvSIG*. Get to know gvSIG Desktop, the Open Source Geographic Information System. 2016. URL: http://www.gvsig.com/en/products/gvsig-desktop (visited on 10/29/2020).

[Ül+06] O Ülgen et al. „Simulation methodology: A practitioner's perspective". In: *Dearborn, MI: University of Michigan* 1 (2006), pp. 7–8.