Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Marek M o d r ý

Study programme: Software Engineering and Management

Specialisation: Intelligent Systems

Title of Bachelor Project: Agent-based Modelling of Urban Crime

Guidelines:

1. Learn about agent-based crime modelling.
2. Get familiar with AgentPolis and A-lite simulation platforms.
3. Propose an agent-based model of urban crime.
4. Implement the proposed model using the AgentPolis platform.
5. Analyse key properties of the implemented model.

Bibliography/Sources:

[1] Miller, J. H.; Page, S. E.: Complex Adaptive Systems: An Introduction to Computational Models of Social Life, Princeton University Press, NJ, 2007.
[2] Bonabeau, E.: Agent-based modeling: methods and techniques for simulating human systems. Proceeding of the National Academy of Sciences 99(3): 7280-7287, 2002.
[3] Malleson, N.; Heppenstall, A. and See, L.: Crime reduction through simulation: An agent--based model of burglary, Computers, Environment and Urban Systems, 34(3), 236-250, May 2010.
[4] Melo, A.; Belchior, M. and Furtado, V.: Analyzing Police Patrol Routes by Simulating the Physical Reorganization of Agents. Multi-agent Based Simulation 2005: 99-114.
[5] Bosse, T. and Gerritsen, C.: Agent-based simulation of the spatial dynamics of crime: on the interplay between criminal hot spots and reputation. Proceedings of Int. Conference on Autonomous Agents and Multi-Agent Systems 2008: 1129-1136.

Bachelor Project Supervisor: Ing. Michal Jakob, Ph.D.

Valid until: the end of the winter semester of academic year 2011/2012

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
L.S.
-4-

prof. Ing. Vladimír Mařík, DrSc.
Head of Department

prof. Ing. Boris Šimák, CSc.
Dean

Prague, February 3, 2011

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Marek Modrý

Studijní program: Softwarové technologie a management

Obor: Inteligentní systémy

Název tématu: Agentní modelování městské kriminality

Pokyny pro vypracování:

1. Seznamte se s problematikou agentního modelování (městské) kriminality.
2. Seznamte se se simulačními platformami AgentPolis a A-lite.
3. Navrhněte agentní model městské kriminality.
4. Model implementuje v prostředí simulační platformy AgentPolis.
5. Analyzujte klíčové vlastnosti implementovaného modelu.

Seznam odborné literatury:

[1] Miller, J. H.; Page, S. E.: Complex Adaptive Systems: An Introduction to Computational Models of Social Life, Princeton University Press, NJ, 2007.
[2] Bonabeau, E.: Agent-based modeling: methods and techniques for simulating human systems. Proceeding of the National Academy of Sciences 99(3): 7280-7287, 2002.
[3] Malleson, N.; Heppenstall, A. and See, L.: Crime reduction through simulation: An agent-based model of burglary, Computers, Environment and Urban Systems, 34(3), 236-250, May 2010.
[4] Melo, A.; Belchior, M. and Furtado, V.: Analyzing Police Patrol Routes by Simulating the Physical Reorganization of Agents. Multi-agent Based Simulation 2005: 99-114.
[5] Bosse, T. and Gerritsen, C.: Agent-based simulation of the spatial dynamics of crime: on the interplay between criminal hot spots and reputation. Proceedings of Int. Conference on Autonomous Agents and Multi-Agent Systems 2008: 1129-1136.

Vedoucí bakalářské práce: Ing. Michal Jakob, Ph.D.

Platnost zadání: do konce zimního semestru 2011/2012

L.S.

prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry

prof. Ing. Boris Šimák, CSc.
děkan

V Praze dne 3. 2. 2011

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

Bachelor Project

Agent-based Modelling of Urban Crime

Marek Modrý

Supervisor: Ing. Michal Jakob, Ph.D

Study Programme: Software Engineering and Management

Field of Study: Intelligent Systems

May 27, 2011

Aknowledgements

Above all, I am truly thankful to my supervisor, Ing. Michal Jakob, Ph.D, whose encouragement, guidance and support from the initial to the final part enabled me to fully understand the subject and write this work.

Next, I am indebted to many of my friends who encouraged me to the completion of the project. And lastly, I would like to thank my family to support me in any respect during the whole time of my studies.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.
I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

In Prague on May 26, 2011 ....

Abstract

Research in the branch of agent-based modelling has experiencing steep growth of the interest in recent years because it allows us to model complicated and heterogeneous systems. In this project, we are focused on urban crime because it is one of the most discussed topics, nowadays. We can emulate the behaviour of policemen and criminals and examine the circumstances of crimes and create possible strategies for crime prevention and persecution.

First, we will explain the basic features of agent-based modelling, and especially modelling of crime. We propose our own model of urban crime, including all behavioural aspects, single interactions and dependencies among system components. After this, we get familiar with A-lite and AgentPolis simulation platforms and we use them for implementation of our previously proposed model. Finally, we perform a few experiments. We will show that our model is correct and then several simulation scenarios are created and simulated and the results of our simulations are discussed.The results show that agent-based crime modelling is very interesting and perspective issue which is definitely worth of future research.

Abstrakt

Výzkum a užívání agentního modelování zažívá v posledních letech velký rozmach. Agentní modelování totiž umožňuje modelovat i takové systémy, které by bylo jinak správně vymodelovat téměř nemožné.V této práci se zabýváme městskou kriminalitou, protože je to stále jedno z nejožehavějších témat každodenního život a díky agentnímu modelování máme nyní možnost vytvořit model systému zločinců a policistů, napodobit jejich chování a v následných simulacích zkoumat okolnosti zločinů, možné strategie potírání kriminality atd.

Na začátku práce uvádíme čtenáře do pojmu agentního modelování kriminality a popisujeme základní vlastnosti. Poté na základě dříve nabytých znalostí navrhneme náš vlastní model městské kriminality, včetně návrhu chování a jednotlivých interakcí mezi komponentami systému. Po seznámení se se simulačními platformami A-lite a AgentPolis je využijeme a implementujeme model, který jsme v předchozí kapitole navrhli. Na závěr provedeme několik experimentů, při nichž prokážeme, že náš model je funkční a dále navrhneme pár různých simulačních scénářů, jejichž zajímavé výsledky zhodnotíme a diskutujeme.Výsledky ukazují, že je agentní modelování městské kriminality rozhodně velmi zajímavým perspektivním tématem, které si zaslouží pozornost.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays, the level of urbanization is continuously increasing all over the world. People move from the countryside or small towns to bigger cities. They want to get better job, better education or just wish to enjoy the modern life in the city. As people move to the cities, the urban area grows bigger and we can find huge amount of people concentrated at one quite small place. And as we know, higher density of people unfortunately creates good opportunities for potential criminal act.

A lot of natural criminals are attracted by such places and even such people who would never steal anything occasionally use the chance to easily gain something valuable. These facts are the reasons why the crime rate in big growing cities is so high and is still increasing.

Wide urban area and high number of criminals mean that there are also a lot of police officers in the streets who try to prevent crime and protect the citizens. Obviously, to make the crime prevention as efficient as possible it is highly desired to carefully choose the right police strategy. There are so many possible strategies, prevention methods etc. that the complexity of choice is very high and it is really difficult to predict all the consequences related to the choice — to pick the right choice is often out of the abilities of an ordinary person.

Besides, when you want to change the strategy or when you want to try a new approach, you have no chance to test it in the real life. The risk of fatal failure and steep growth of the crime rate is so high that you can hardly dare to apply any new strategy in the streets of any town without previous testing.

There are usually two ways how to analyze some criminological issue. Either you need to use the help of a criminological expert and let him work for a long time on the analysis or it is necessary to use an auxiliary software utility. Such software analyzer can save our time, it is usually more accurate, and it particularly reduces user's requirements for a professional education in the criminology branch. It means that almost anybody can easily use the software, get useful results or change analysis parameters and almost instantly get new refreshed results.

There are many ways how we can analyze and study criminality, the crime prevention strategies, the crime rate trends etc. Computer simulation of urban crime is one of them. Simulation provide us with the possibility to easily check the consequences of any change we would make in our environment. [2]

The core of any similar system is based on a model of the environment and of all the things which can have any influence on the environment. It means that we have to provide the system with set of rules, parameters and other important information according to which the system(or simulation) will run.

This very model and its creation (called modelling) is one of the main topics of this work.

First, scientific modelling is described and main intentions for using it are listed in Chapter 2. Basic modelling approaches are named and then compared to each other. Significant part is concentrated on introduction of agent-based approach and the reasons why it was chosen for this project are explained.

Further in Chapter 2, background for agent-base crime modelling is described. Fundamentals of modelling process and the reasons for use of scientific modelling are named. Two basic modelling approaches are examined and several advantages and disadvantages of agent-based modelling approach are stated. At the end of Chapter 2, we provide the reader with an idea what for and how to use agent-based crime modelling. Finally, couple of related projects are referred.

Next, in Chapter 3, we propose a model that could be used for agent-based simulation with concentration on urban crime and all its essential parts. We analyze and propose a model of fundamental parts of the simulation. An urban environment is modelled, as well as, criminals and policemen who are the crucial part of the model. Their behaviour and rules of the interactions between criminals and policemen were examined and described.

Chapter 4 is narrowly linked with the previous section. It talks about the particular implementation of previously proposed model. In this chapter, descriptions of an architecture and features of our implementation are given. Moreover, A-lite simulation platform and AgentPolis simulation framework are shortly briefed and the dependencies are interpreted.

Chapter 5 is following. In this chapter, we analyse key properties of the implemented model. We proposed a few possible simulation scenarios and run a simulation to observe dependencies, relations and interactions between two main agents group — criminals and policemen. We perform a few experiments and provide the reader with the results. At the end we discuss the results and try to examine the context.

Finally, in Chapter 6 we recapitulate this project, its aims and its results. We also provide the list of possible future extensions and suggestions for future work on this topic.

Chapter 2

Background

As we mentioned in Chapter 1 such situations can occur when we need to find a solution for some problem and we can't afford to perform any experiments in the real world. It can be too expensive, too risky or just impossible. For instance, when you want to find something out about epidemic diffusion, you can't just let some virus to attack people and then trace how it spreads. You have to try it on an artificially created world. And this is the very reason why we need to create a model of the system that we want to examine.

2.1 Modelling approaches

There are several possible approaches to modelling of various systems. According to [3], modelling, as we call the procedure of model creation, can be divided into two main groups. The first one is analytical modeling. Analytical modeling (sometimes called even static) depends particularly on the input and we are usually able to get the results immediately. The second approach is simulation modelling. Difference between these two approaches is in the fact that the simulation model is provided with the information about how the system will change in the future when given the current state. In this case, we can't get the results immediately. We have to wait for sufficient number of simulation steps to be done.

We have used agent-based modelling in our project. However, to show you the reason of this choice, it is necessary to explain the basic categories of various modelling techniques. Similarly to the previously mentioned division of modeling approaches, we can name two ways how to represent the simulation model. There are more of them but those two are the most important to describe the pros and cons of the approach which was used in this work.

First, equation-based modelling approach which is used really frequently. It uses mainly a set of mathematical equations (usually algebraic, particularly differential equations) on the aggregative basis and input variables. [16] Those equations describe the modelled system and define how the system will change in the next time steps. This approach is very effective and we can get the results again almost immediately. Although it is still often used, we can see that nowadays there are new approaches that slowly substitutes the one mentioned as a first.

As the second, agent-based modelling approach which has became popular only in recent years. More about this approach is written in Section 2.2.

2.2 Agent-based modelling

Agent-based modelling which is used in this work is based on the usage of independent decentralized interactive units called agents. We often try to model the system containing thousands of agents and it could be very computationally demanding. It is the reason why the agent-based modelling wasn't used much in the past. However, recent rapid technological, computational and data storage advancement allowed us to use the agent-based applications in a greater extent.

Agents are autonomous decentralized independent small units placed in an environment. The environment is artificially modeled world in which the agents exist. Agents are capable of making their own independent decisions according to their characteristics, rules and decision-making capabilities which are given by the model. Every agent can interact with other agents (in the case, it is defined by the model). Charles and Macal claim in [10] that agent should be also capable of learning from its experience and adapting according to changes of the environment. Thanks to independence of agent units we can model heterogeneous populations of agents. While with usual differential equations, it is very complicated to represent such populations.

The fundamental purpose of an agent is to exist in an environment and to somehow interact and react on signals incoming into their sensors. Demands on the behaviour and usage of agents can be very various. Agents can have memory, cooperate with each other, plan and schedule their future steps in advance etc. (so-called rational agent) or it can be more or less simple without any advanced intelligence and just reacts on the stimulants from the environment (so-called reactive agent) [14, 9]. It depends on the fact what we use the agent-based model for. Environment can be different in various ways as well. The environment can be both, changing itself in the time, continuous or it can be also partially hidden for the sensors of agents. It can be discrete as well, still the same without any changes in time and provide agents with all information about current state. More about agents, for example, in [2, 3, 10, 14].

In comparison to other approaches, agent-based modelling has many advantages. It provides relatively high flexibility of the model. Individual parts of the model are quite autonomous and can be easily modularized. It is also easy to add any new feature, new behavior, new environment variables etc. [14] Besides, you can easily model all interdependencies and even models of large-scale micro-simulations are well transparent and natural. [10] claims that thanks to both, potential diversity of agents and all options that are offered by agent-based modelling, you can create such model that will be able to answer almost any real world question. In [2], the author writes even more about the pros and cons. For instance, another advantage which we haven't written about yet, but which is also quite important is that you can discover emergent phenomena which you wouldn't expected and which you wouldn't discover when using other approaches.

Nevertheless, you should take into account that there are a few disadvantages as well. Similarly to any other modelling approach you have to decide what is the purpose of your model. There is not anything like "general purpose model". And the decision, what the purpose of the model is, isn't always easy. Another thing is that once we have created the model it is extremely important to calibrate the variables used in the model (or simulation). Otherwise you can get completely useless results. And as written above, one of the worst disadvantages is that application of agent-based model can be computationally very demanding for large-scale systems.

Examples comparing agent-based modelling to equation-based modelling using the differential equations can be seen in [3, 6].

2.3 Agent crime modelling

Power of agent-based modelling is quite obvious. And it makes sense that people try to use such a powerful instrument for their purposes. There are many branches where agent-based modelling could be used — medicine, economics, transportation, safety engineering, social interactions research etc. In addition, it can be very useful for simulation of urban life. Agent-based modelling provide you with the possibility to model almost any behavior and interactions of the citizens. Possible diversity is very helpful when human behaviour is modelled because each person behaves a little differently compared to others.

Related works on this topic are usually focused on the movement of all the movable agents (people, cars, public transport, . . .) and their interactions. There is relatively large number of works related to usual urban life modelling and consecutive simulation. However, there is still huge space for future research and development of new approaches and techniques as stated in [1] by Benenson and Torrens.

In addition to the previously mentioned and many other branches, there is one very interesting and perspective issue which is worth of attention. And the issue is criminality. As we discussed in Chapter 1 and at the beginning of this section, there is a lot of reasons for crime modelling and simulation. It is good way how to find the best strategy for the crime prevention and persecution. By changing a few parameters we can create various scenarios of urban criminality and observe all the interactions in our system.

Just for your imagination, couple of scenarios and adjustments which can be easily simulated will be named:

Police patrolling strategy The spatial distribution strategy of police forces could be adjusted and the most effective strategy would be searched according to the results.

Victims choice strategy of criminals The way how the criminals choose their new victim and the place could be changed. Then, such settings would be found which would cause that the crime rate would increase. Then the reason for vulnerability of police and citizens would be searched. If the reason was known then the behaviour of policemen and citizens could be changed to decrease the crime rate again.

Police officers using bicycles We can adjust the value of police officers' walking speed. By this adjustment, we can simulate the situation when the police officers are provided with bicycles (or other similar way of movement). Then the effects of speed adjustments could be observed.

Obvious visibility of the police forces The criminals' awareness of police could be adjusted. We can observe the differences between the case in which policemen are obviously visible and the case in which policemen are hidden. This can be modeled by adding the probability of ignoring the near-standing policemen when the criminal wants to commit the crime. The second way how to model this scenario is by decreasing the parameter determining the range of sight of criminals when they look for a policeman.

Nevertheless, what is the reason for using the agent-based model for crime modelling? As it is written in [7], all criminal events result from the combination of offender motivation, opportunity, target characteristics and an environmental backcloth at a particular point in space-time. In addition, all agents which are somehow involved in the criminal event are influenced by a sequence of historical events (e.g. if their victim decided one hour ago to use the same street). Those historical events are also influenced by the constraints of the environment. From this, we can deduce that there is high emphasis placed on the history of all the subjects.

We have already discussed the differences in modelling approaches. Therefore, it is obvious why we chose the agent-based modelling. It is important to follow all the agents separately, as individuals. Otherwise, you are not able to efficiently model the opportunity which arose from the combination of an individual agent, an environment, space-time and all circumstances.

In addition, if you want to analyze history and the reasons for committing a crime, you have to know what happened before. You need to have information about previous and current states of an agent, an environment and other involved components. And in the case of analytical modeling, you do not have proper information. The differential equations can hardly express all relations, states and history of an agent. In comparison, when you use agent-based modelling you can observe actions, states of each single component of the model (or simulation) and related circumstances.

[7] continues with thought that you can determine crime event by the sequence of actions which influenced the criminal in the past. It means that you can identify the moment when the criminal will commit the crime with high probability. In other words, you can find the templates in the criminals behavior. Then, those templates of actions can determine when the criminal event should happen in the simulation. Of course, this can only hardly be modelled while using the analytical modelling.

2.4 Related work

There is a lot of different issues related to urban simulations which can be modelled. For example MATSim[1] provides you with a toolbox to implement large-scale agent-based transport simulations. It offers you the tools for mobility-simulation, for analyzing the demands of the transport, for re-planning and methods to analyze the outputs of the simulation runs. Similar tool is Simulation of Urban Mobility (abbreviated as SUMO)[2]. It provides you with a simulation toolbox for multi-modal traffic simulation. It is focused mainly on the analysis of transport demands in accordance to road network.

As our project deals with agent-based modelling of urban crime, our interest is concentrated mainly on agent-based crime modelling. There are a few related projects which should be noticed.

[1]Multi-Agent Transport Simulation - http://www.matsim.org/

[2]http://sumo.sourceforge.net/

2.4.1 Spatial dynamics of crime: Hot spots and reputation

First of these projects is [4]. This work is focused on the urban crime simulation. It deals with the relations between spatial dynamics and the criminal hot spots creation. Criminal hot spot is the place where the risk of crime is higher than at other places. It usually means that there were recently many crimes committed at the same place. It is important to mention especially one thing — the fact that authors of [4] used predicate logic to describe the states and the changes of environment. Because this is the reason why their work differs from others.

To model the system, they use LEADSTO[3] language. LEADSTO is special predicate logic language developed by the authors of [4]. It was developed especially for the purposes of simulation modelling. LEADSTO was derived from TTL [4]. TTL was used because it satisfies all requirements of authors. Dynamic processes can be modelled by specifying the direct dependencies between state properties in successive states. Both, logical and numerical aspects can be modelled. Moreover, as authors of [5] stated, it is a declarative order-sorted temporal language.

As mentioned above, authors of [4] created a model of criminality and of the creation of hot spots depending on the reputation and attractiveness of each place in the environment. The authors presented the environment as a graph of nodes connected with edges. We can imagine that as the potential crime places and the roads which connect them together.

The environment and its states change in every time iteration. In every step, all the agents move and the simulation proceeds all the actions performed during this very step.

All agents move considering the reputation of all the places. Agents are provided with information about whole environment. It means that environment is fully observable. The reputation assessment runs every step. There are two different types of reputation values. The first one is given by the count of crimes which were successfully committed at the node and the second one is given by the amount of criminals who were arrested by the guardians at the node. Agents in the model are divided into three groups:

passers-by They are victims of the crimes. However, when there is a guardian at the same position, passer-by can't become a victim. Passers-by move to the most attractive position. Such places are attractive for passers-by, where the minimum amount of crimes were committed and the minimum amount of criminals were arrested.

guardians They try to protect the passers-by. Guardians move to places where the number of committed crimes is high because they want to prevent passers-by from becoming the victims of future crimes.

criminals Agents who commit crimes. They try to find the places where the amount of successfully committed crimes is high and, at the same time, the amount of arrested criminals is low.

One of widely discussed questions in agent-based crime modelling is how we should model the interaction between the criminals and the guardians. What should happen when

[3] A Language and Environment for Analysis of Dynamics by SimulaTiOn
[4] Temporal Trace Language

the criminal is arrested? Or what should happen to passers-by when they become a victim of a crime? It is very difficult to describe the subsequent changes of the agents' state when they are involved in the crime.

In [4], when a criminal successfully commits a crime, he is marked as "wanted" person for 4 following simulation steps. If he meets the guardian in those 4 steps he is arrested. If he does not meet any guardian he becomes anonymous criminal who can't be recognized and he starts looking for next victim. Authors of this discussed work observe how hot spots change the position as the time goes and they observe also how changes in simulation parameters can influence the simulation results. The authors use small-scale simulation with only few agents and few nodes (only tens of units). During experiments, they were changing the settings stepwise and then examined the changes of the simulation results.

2.4.2 Agent-based model of burglary

The second related work which is definitely worth of noticing is [13]. As authors wrote, the work is different in the approach to the agent model. They do not use classical theoretical BDI model (where BDI stands for Beliefs, Desires and Intentions). They implemented PECS model because the authors believe that it is better for the purpose of their work. PECS abbreviation stands for Physical conditions, Emotional states, Cognitive capabilities, Social status.

Agents do not behave only on the basis of rationality and beforehand given rules. Their behavior is driven by the basic feelings and instincts which are described by simple time-dependent functions. Therefore, the agent does not decide to go to work because of the rule saying that the agent should leave his flat at 8 o'clock in the morning and set off to his job. In PECS model, the agent sets off to his job because his desire for earning the money overcomes the level of the need for having a rest at his own home.

For simplicity, agents in the model only work and have a rest at their home. Environment is a grid of cells representing the individual positions. Each cell represents either, the road which is used by agents for movement, or the building where the agents live or work, or just the blank cell which is inaccessible. It is important to mention that values determining the security level and evaluation of the worth of the building are assigned to every building in environment.

The day cycle of ordinary citizens is modeled in the way that people sleep over the night and they work over the day. They just move from one building where they live to other building where they work.

The second type of agents in [13] are the criminals. The difference between ordinary citizens and criminals is that the criminals do not have any full-time job. Hence, they are not able to earn enough money to satisfy their desire for money and they are made to commit the crime in order to substitute the source of income. From time to time, criminals can be assigned to part-time job. They get part time jobs because the authors wanted to control the lack of money of the criminals. If the criminal satisfies his desire almost completely he does not have to commit the crimes so often. On the other hand, if he lacks a lot of money (without any part time job) he has to commit many crimes to satisfy his needs. Crimes in [13] are focused on the residential burglaries.

These above mentioned part time jobs have one more purpose. When the criminal goes down the streets towards the part time job place, he remembers all the buildings he has seen during his movement. The criminal remembers all the buildings with all their security and worth values.

The criminal chooses the building to burgle from his cognitive map which was created during his travelling. The roulette choice based on the security and worth of the building is used for the choice from the cognitive map. In addition, when agent goes through the city towards the chosen building he checks if there is any other building with higher attractiveness along his path. When the agent reaches the crime commitment place, he checks if there is anybody in the building. In the case that the building is empty he can burgle it. Otherwise, he leaves the place and starts the whole commitment sequence from the beginning.

Authors of the work experimented on the environment which was composed of 41x31 cells. There were 300 agents in the environment, 5 percent of them were criminals. Authors used the knowledge from [20] and noted that most burglars return to previously burgled properties. They modeled this fact in the way that they temporarily decreased the security of the recently burgled buildings.

They used the model we have written about to examine the effects of different crime prevention strategies. They compared two main crime prevention campaign types. The campaigns are supposed to decrease the crime rate. The first campaign is aimed to cover all the citizens in the city but with the lower efficiency. The second campaign is aimed only on the small part of the city but the efficiency is much higher. They simulated those campaigns by temporary increase of the security level value of the buildings.

Further experiments examined the influence of the spatial distribution of population on the movement of the burglars. Authors simulated different society layers on the basis of different security and worth values of the buildings where people live. What's more, authors tested the truthfulness of a few criminological claims. For instance, if it is true that the higher is the expected value of a loot the longer path the criminals will to travel (noted by [18]).

Chapter 3

Proposed model

As we mentioned in Chapter 2, agent-based modelling was used in this project. We use this modelling approach for above listed reasons.

Important thing to consider when you model a system is the level of abstraction of your model. It is noted in [3] that it is important to decide what should be involved in the model and what shouldn't. When you model a physical system it is understandable that you involve all the physical attributes of the components. Whereas when you model the system which is supposed to show how the car accidents are related to racial composition of the local population, there is no need to model height, width or colour of eyes of the agents.

For example, there are residential burglaries simulated in [13]. Therefore, authors focus on modelling of security and worth values of the buildings. Whereas, for example, roof colour is not taken into account. In comparison, there are no buildings modelled at all in [4].

It is advised to keep the model on a reasonable level of abstraction. If you involve more components than it is necessary, the model becomes nontransparent and even confusing. You can lose the capability of recognizing the right origins of individual events and it will be difficult to examine the environment and all related components. Moreover, it complicates adjustments of a model or related simulation. Thereby we would lose one of the main advantages of agent-based modelling.

Author of [17] noted that it is desired to understand the difference between meanings of the words replica and model. In his opinion it is not necessary to create a replica of a system especially when we create the model for research purposes. We usually examine only some limited phenomenon, hence it is even advised to create simplified model which is not interfered with other unrelated influences. However, you have to decompose all the problem layers really thoroughly. Good example of simplified model can be seen in [4].

3.1 Purpose of model

Criminology is scientific study with wide range of interest. Therefore there is a lot of interesting issues you can examine by simulation. There are many different victim types which you can commit the crime on (vehicles, residents, citizens, companies, ...). There are also a lot of different ways how you can offend a victim (pickpocket, assault, burgle, ...). Furthermore, when you know which kind of crime you would like to model, you have to decide what you will focus on. There are several consequences which you can concentrate on — When was the crime committed? Where was the crime committed? Who was the victim? How does it come that the criminal decided to commit the crime?

We didn't create the model with intention to examine neither the crime opportunity theory[1] and its consequences nor the reasons for committing the crime nor even any criminal habits. There are works which has already done that. Apart from them we focused on the relations between criminals and police forces which try to prevent and persecute the crime.

We are mainly interested in the influence of different crime prevention strategies on the crime commitment success rate. Following changes can also be counted among changes in crime prevention strategy. For instance, number of policemen could be increased, their speed could be increased, police behaviour could be changed etc. While adjusting parameters of the model and the simulation, we would like to observe the changes in the criminals success rates and even influence of police forces on the spatial distribution of criminals. Thus, we build a model which meets both following requirements — on the one hand, the model involves all the features which are essential for the simulation and on the other hand it does not handle with anything that could make our simulation nontransparent.

3.2 General overview

What is described by the model we have created? There is an environment (let's call it city). And there are criminals and policemen who walk around the city. Criminals travel around the city and commit the crimes at such places which they find to be the most promising for success of their intentions. Similarly, there are policemen who are distributed all around the city, and they try to prevent or interrupt as many crime events as possible.

Criminals want to maximize their average crime commitment success rate. On contrary, policemen want to minimize the success of the criminals and to maximize the success of each patrol (the success of policeman is when he interrupts or prevents a crime). See Figure 3.2 for illustration.

As we agree with the opinion of [7] about the fact that crime events are significantly influenced by the historical trajectory of each component involved in the criminal act, we decided to provide both criminals and policemen with the memory and learning ability to be able to behave according to their history.

Proposed model does not describe any directly defined needs or desires. In the model, there isn't any "desire function" similar to the one in [13]. In the model we propose that agents are driven by a given cycle of states. Every agent goes through sequence of successive states. Every action/decision(finishing patrol, committing crime etc.) then depends on the current state and, of course, on the states which the agent was in before. The sequence cycle repeats in never-ending loop. If agent finishes the sequence cycle he immediately starts with a new one. This approach reminds us of [7] where the theory of templates is described (and template can be interpreted as sequence of actions which repeats before each crime event).

[1] Opportunity theory deals with the interaction between the victim and potential offender. It examines the relations, circumstances and reasons.[8]

Figure 3.1: Illustration of the policeman on the patrol and the criminal who is committing the crime

3.3 Environment

Environment in our model is quite simple. It represents a city (for simplicity, in most cases, the expression "city" will be used instead of "environment" in the following text). The city consists only from positions in the city where the agent can perform actions and roads which are used for connection of previously mentioned positions. However, it isn't possible to perform any action in the middle of road. The inhabitants of the city are only criminals and policemen.

There are no means of public transport in the city or any other possibilities how the movement could be accelerated. There are also no buildings in the city. As only criminals and policemen are involved in the model, there is no need to model the buildings because they do not interact with the buildings anyhow.

In the environment the time goes as at any other place. Whenever we will write about the time in the following sections we will mean the environment time (unless stated otherwise). However, it is necessary to mention that the time isn't the main subject of interest in this project. Thus, when we will write about the results of experiments, it is not possible to compare them with the real world statistics from the view of time. It is obvious that in real world, the time influences criminality a lot. Crime rate depends also on the daylight which is related to the time. Neither of those fact are modelled.

To use the time information correctly in our model, we would need a lot of real world data for calibration of our model. Unfortunately, it wouldn't be possible to easily acquire it. Therefore, the exact relation between time and the environment was omitted.

3.3.0.1 Sectors

According to our own experience it is usual that people do not remember exact places of their experiences. They rather assign a wider area to the particular experience in their minds.

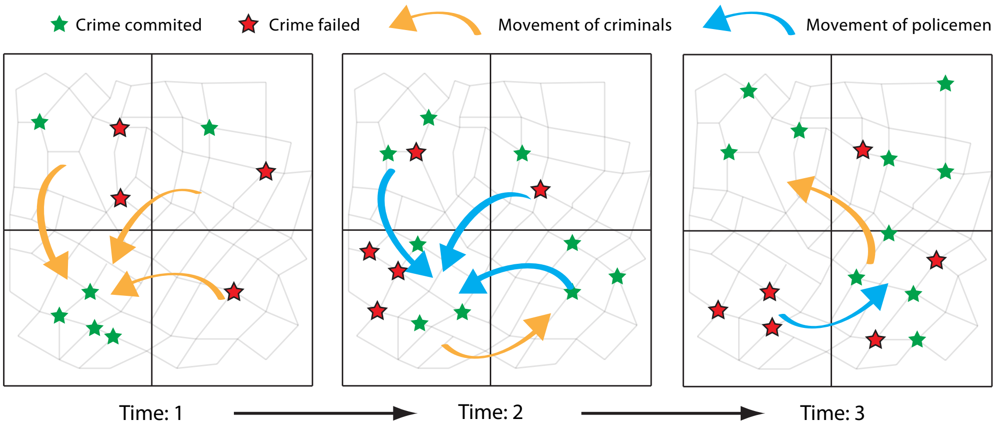


Figure 3.2: Scheme of agents' movements among sectors

For this reason, when agents are deciding about something spatial their considerations are running on the basis of wider areas. Whole city is divided into several square sectors. Those sectors represent those wider areas in the minds of agents. For instance, when an agent wants to remember his experience with the current place he does not remember the current exact position (e.g. crossing of Oxford St. and Regent St.) but he remembers that the experience has occurred at current sector (e.g. 1 square km at West End).

We can explain it further on one more example — if a criminal remembered all the experiences in a connection to the exact positions it would take unimaginably long time to build a good cognitive map. On the contrary, if a criminal remembered that his success rate in Hoboken quarter is about 75% and in Soho about 15% it would be definitely more useful than the case with exact positions. In this particular case, he would probably decide to commit the next crime in Hoboken.

Finally, it should be noted that when agent chooses a place to perform any action — he chooses a sector, not exact position. The exact position is then chosen randomly. More about particular implementation is written in Chapter 4.

3.3.1 Crime event

It is important to define what the crime actually means in our model. First, we should note that no victims are involved in crime events as we model it. The project is not supposed to examine neither the influence of the criminals on the victims nor the reactions of victims on the changes of local crime commitment rate. Furthermore, if we wanted to add the interaction with the potential victims it could have made our model less transparent. (This is also the reason why there is no possibility to use public transport or private cars for the similar reason.)

Crime event in our model is the moment when criminal decides to commit a crime. The crime can be committed at any available position. It can be spotted or even interrupted by a policeman or it can be successfully finished without any notice from the side of police. However, there is also the possibility that the criminal wants to commit a crime but he can't start due to the fact that a policeman is near to him. More about particular behaviour of agents will be written in following parts.

3.4 Agents

There are no ordinary citizens in the city. There are only policemen and criminals. Agent can't use any movable vehicles or anything that could make their movement faster — neither public transport means nor private car nor bicycle. When an agent decides to travel from one place to another, it is necessary to go on foot.

Every agent is born at the beginning of the simulation. However, there is short randomly chosen time interval which determines the exact time of birth. The randomly chosen interval is added because otherwise some impulsive waves of actions could appear at the beginning and we know even from the real world that people do not act at once and the times when they decide to do something diverse.

When the agent is born he starts to behave according to the fact whether he is a criminal or a policeman. No agent needs neither to sleep nor to eat nor anyhow satisfy his human needs. The only thing they do is that they carry out their functions. It means that they either commit the crimes or patrol and try to prevent the crimes.

The absolute number and the ratio of criminals to policemen can vary. In following sections, we will examine the influence of changes in the numbers of criminals and policemen.

3.4.0.1 Learning from experience

As mentioned above, all agents are provided with the memory and ability to learn from experiences. This learning issue could be classified as reinforcement learning because agents try to maximize the reward they earn by the choice of particular sector for the crime commitment. In this project, rewards are chosen only from a set $\{0, 1\}$. Agents then decide according to their estimated probabilities of success rate in each single sector. Multi-armed Bandit algorithm[19, 11] was used as a base for our modified algorithm because they deal with similar problem. Particularly ε-greedy algorithm was chosen as the base for modifications.

In short about multi-armed bandit problem. Imagine a slot machine with three levers. Multi-armed bandit problem deals with the problem when you want to choose such lever which would maximalize your reward. The main problem is that you should choose the lever with the highest possible reward and you should also explore rewards related to other levers and their changes at the same time. So you solve two problems at once — how to estimate the rewards related to the levers and how to balance between exploitation and exploration.

Now, something about the mechanism of our memory will be written. When an experience is recorded, it is added to the list of experiences in agent's memory. However, as it is generally known, one can't remember everything forever. And it is apparent that after some time the experience loses validity. For instance, if one remembers that the crime rate at 1st

Avenue was high yesterday, it is highly possible that the crime rate will be high today, too. On contrary, if one remembers that the crime rate at 1st Avenue was high 2 years ago, it will not be likely the same today. For this reason, experiences older than given time (initially set to 5 days) are removed from the memory. In addition, the experiences lose their importance as the time goes. The function describing the decrease of importance of the experiences is shaped as the cosine curve(see Figure 4.8).

The record consists of three main parts — first is the current time (when the recorded event happened), spatial information (identification of a sector where the event happened) and finally the particular experience. An experience is represented by binary value (0 or 1) depending on the fact if the experience was positive or negative. The word "positive" does not mean that the experience was good but it implies that the place should attract more attention in the future. For example, a policeman would record a positive experience in the case that he interrupted the crime commitment. And again when a policeman finds out that the sector is secure (no crime was spotted during his patrol) he will record a negative experience.

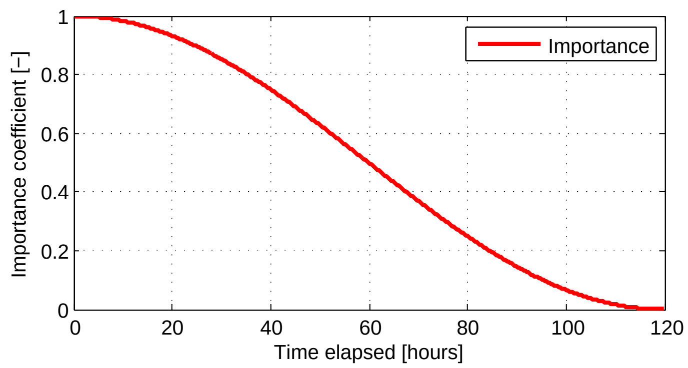


Figure 3.3: Importance function of agents' memory

3.4.1 Criminals

Criminals are agents that travel around the city and commit crimes. The principles of a cycle(sequence of actions) which the agents are driven by were mentioned above.

Now we will describe one complete cycle which loops for entire life of a criminal. Whole sequence begins with a choice of the area where the criminal plans to commit the next crime.

3.4.1.1 Choice of the next crime place

There are several circumstances which can influence choice of the next crime place. In the very beginning of the simulation there is a set of warm-up crimes during which the criminal randomly choose the sector regardless of any previous experience. If the criminal would choose a sector according to his previous experience right at the beginning it would be either impossible(e.g. no experience is in the memory) or misleading.

After a few warm-up crimes the criminal is capable of qualified decisions. However, there is still a small probability that the criminal decides randomly (initially 25%). This feature should ensure that criminals keep on exploring the city and new crime commitment opportunities. If a criminal does not decide neither to explore the map nor to choose the sector randomly in the warm-up phase he chooses one of the sectors where the crime commitment success rate is on the level of one of the highest values. It is not sure that the chosen sector is the one with the highest crime commitment success rate. The probability of being chosen is divided among a few best sectors(initially 10 percent) according to their estimated success rate. If the best one was always chosen the model wouldn't be variable enough. In experiments where criminals share the memory, choosing always the best sector would cause massive movements of criminals to the best sector without any variety.

When the criminal chooses the next crime position, he sets off to the chosen place.

3.4.1.2 Committing crime

At this time, the criminal is already at the place which was chosen in the previous step(in Section 3.4.1.1). It is not necessary to look for any victims because as we noted above, criminals in our model commit the crimes independently on victims. The criminal can actually start the crime commitment wherever he wants.

The criminal is prepared for the crime commitment. At this point, criminal checks if no policemen are around. "Be around" means that policeman is at the position which is nearer than given radius (radius is set according to the simulation settings, initially 50 meters). If no policeman is around the criminal can start. However, there is also a probability that the criminal does not notice the policeman. If the chance determines that the criminal does not notice the policeman the criminal does not even look around himself and he immediately starts committing a crime. If there is a policeman somewhere nearby the criminal records this experience in his memory and starts the sequence from the beginning (new choice of crime commitment place, travelling, etc.).

If there is no policeman around the criminal starts committing the crime. The criminal is committing the crime for a time randomly chosen from given interval. There are two ways how the crime can end. Either the crime successfully ends after the given time has elapsed or the crime is interrupted by a policeman. Both variants will be described in the following sections.

3.4.1.3 Successful crime commitment

A crime is considered to be "successfully committed" when the criminal hasn't been disturbed by any policeman for the whole duration of the crime. In the moment when the crime has been successfully finished the criminal records an experience into his memory. It should be reminded that three main parts of an experience record are, the current time when the crime finished, sector where the crime was committed and especially the kind of experience(positive or negative).

3.4.1.4 Failed crime commitment

A crime is considered to be "failed" when the criminal was interrupted while committing the crime. Crime commitment fails when there is a policeman and criminal at the same position at the same time and the policeman decides to interrupt the criminal that is committing the crime. Note that criminal can be recognized only when he is committing a crime. Otherwise, a criminal can't be recognized by a policeman and the policeman does not take care about the criminal at all. If a criminal is interrupted during the commitment (the crime commitment is failed) the criminal records this experience in his memory. The record is very similar to the record about the successful crime. The only difference is that in this case the negative experience is recorded.

3.4.1.5 Between crime events

Criminal starts new sequence from the beginning after his attempt to commit a crime regardless of the fact if the crime was committed successfully or not. (new choice of crime commitment place, travelling, etc.)

When the crime commitment is finished, the next sequence of actions starts. Only constraint is that a criminal can't commit a crime more than once per a given time. This is checked always before the start of the next crime when the criminal is at his chosen place. If needed, the criminal waits until he fulfils time interval condition. The given time is determined by parameter settings (initially 20 minutes).

3.4.2 Policemen

The purpose of policemen is to prevent the crime by their presence and to interrupt as many crimes as possible. Although, their life cycle is quite similar to the life cycle of criminals, there are some important differences.

3.4.2.1 Police memory

There is one important feature in the model — all the policemen share one common memory. It means that all the experiences are recorded in the same memory and each policeman can learn from experience of all his colleagues.

The memory is processed every few minutes of the simulation time. The table evaluating crime-exposure for each single sector is built according to the experiences in memory. Exposure of sector i is calculated as follows

$$exposure_i = \frac{\sum_{k=1}^{n} (importanceCoef(experience_k) \times experience_k)}{\sum_{k=1}^{n} importanceCoef(experience_k)}, \tag{3.1}$$

where $experience$ is a set of all valid experiences related to sector i, n is size the of an $experience$ and $importanceCoef_k$ is importance multiplier of $experience_k$ derived from the time when the experience happened and the current time. Experience importance multiplier is calculated from importance function which was mentioned above (see Section 3.4.0.1). Value of $exposure_i$ is calculated for each sector.

3.4.2.2 Central control of police forces

When the experiences are analyzed a table of police forces distribution is created. The police forces distribution table shows how many policemen should be allocated for each sector in the environment. Number of policemen which should be allocated for particular sector is calculated as follows

$$allocation_i = round(\frac{exposure_i}{\sum_{k=1}^{n} exposure_k} \times numberOfPolicemen), \tag{3.2}$$

where a normalized value of $exposure_i$ is used. Let's assign

$$normalizedExposure_i = \frac{exposure_i}{\sum_{k=1}^{n} exposure_k}. \tag{3.3}$$

$\sum_{i=1}^{n} normalizedExposure_i$ equals 1 and $normalizedExposure_i$ is multiplied by the total number of available policemen. This formula gives us an allocation distribution of policemen for all the sectors.

By influence of rounding there is not certainty that

$$\sum_{i=1}^{n} allocation_i = numberOfPolicemen. \tag{3.4}$$

Therefore, if

$$\sum_{i=1}^{n} allocation_i \geq numberOfPolicemen, \tag{3.5}$$

then nothing matters because every policeman could be assigned to a sector and the fact that a few sectors wouldn't be fully covered wouldn't influence the results anyhow noticeably. Only a few units would miss. However, when

$$\sum_{i=1}^{n} allocation_i < numberOfPolicemen, \tag{3.6}$$

then $(numberOfPolicemen - \sum_{i=1}^{n} allocation_i)$ more policemen are allocated to randomly chosen sector (each policeman is allocated separately).

Except of allocation table, there is another table which keeps information about the numbers of policemen who are currently patrolling the particular sector.

3.4.2.3 Choice of patrol place

At the beginning of the cycle, a policeman chooses the next place where he will patrol. How does he do that? The policeman compares allocation table to the table with current police forces distribution and he finds a set of sectors which fulfil following conditions. First, there are actually less policemen than it is necessary. Second, it is the closest sector in comparison to the current sector where the policeman stands. As there can be more sectors which are the closest, exact sector is then chosen randomly from the set of sectors which fulfil given conditions.

When a sector is chosen, exact starting position of the patrol route within the sector is chosen randomly and the policeman is registered to the table which keeps the information about numbers of policemen who are currently in particular sectors.

3.4.2.4 Patrolling

When a policeman arrives to the chosen position he starts patrolling. Patrolling is 30 minutes (of simulation time) long period. During that period policeman is randomly searching through the positions which are within the sector. It means that policeman can't move out of the sector where he is patrolling. Always when policeman moves to next position (completes the path from one position to another) he checks if there is any crime being committed in the surroundings. Surroundings are defined by a circle with radius initially set to 120 meters (see Figure 3.4 for illustration). In the case there is only one position available in the sector, a policeman stands at the same position for the whole time and he checks the surroundings every 2 minutes.

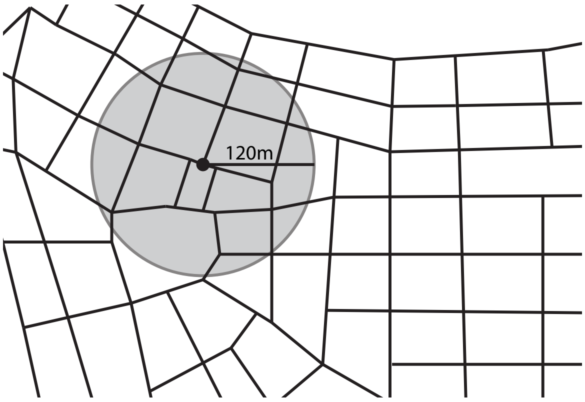


Figure 3.4: Illustration of surrounding radius

3.4.2.5 Crime spotted

When a policeman spots the criminal who is committing a crime he remembers the position of the spotted crime. The policeman immediately sets off to the position where he has spotted the crime. Of course, it takes some time to get there. When the policeman arrives at the position of the crime which was spotted, the policeman checks if the crime is still being committed or if the criminal has already left.

If the criminal is still at the same position and he is still committing the crime, he is interrupted by the policeman who spotted the crime before. When the crime commitment is interrupted the policeman records the positive experience to his memory. Positive experience would be recorded, therefore, the police should pay more attention to current sector in the future. In the case the criminal has already left no experience is recorded. The reason for this is that the criminal could have been interrupted by somebody else already or more policemen could have seen the criminal at once. Then the same experience would be recorded multiple times and it isn't right. This is solved other way. When the crime is successfully committed and a criminal has left before any policemen arrive then the positive experience is recorded to the police memory. It can be imagined as a crime which was reported to the police by imaginary citizens. The experience related to this crime is recorded only once.

Patrolling can finish after 30 minutes without any crime spotted during the whole patrolling period. Then, the negative experience is recorded. It means that less policemen will be attracted by current sector in the future.

3.4.2.6 Patrol finished

Policeman who spotted a crime, arrived at the crime place and somehow reacted on the current state of the spotted crime then he continues in his patrol until he finishes 30-minute period.

When a policeman finishes the patrolling period he deregisters from the table of current police force distribution and he starts whole cycle from the beginning. The policeman checks if he is needed at the current node for next patrolling period. If he is needed he continues with his patrolling at the same sector. If not he chooses a new sector to guard.

3.5 Conclusion

We have proposed a model for agent-based simulation of urban crime. We have defined how the crime event would look like and how would interact all the components. Model of agents' behaviour was proposed and we described how would their life cycle look like. In Chapter 4, we will use the model we have proposed and our own crime model will be implemented.

Chapter 4

Implementation

We have proposed a model in previous Chapter 3. The model is partially based on information which we noted especially in Chapter 2. To have the possibility of verifying, examining and experimenting, we have used AgentPolis simulation framework and extended it for our purposes. The extended framework implements proposed model and allows to perform all the above mentioned actions. It provides us with the opportunity to simulate the life cycles of criminals and policemen. We can observe their behaviour, easily change the settings of simulation parameters and, furthermore, we can implement many interesting scenarios without any complicated difficulties.

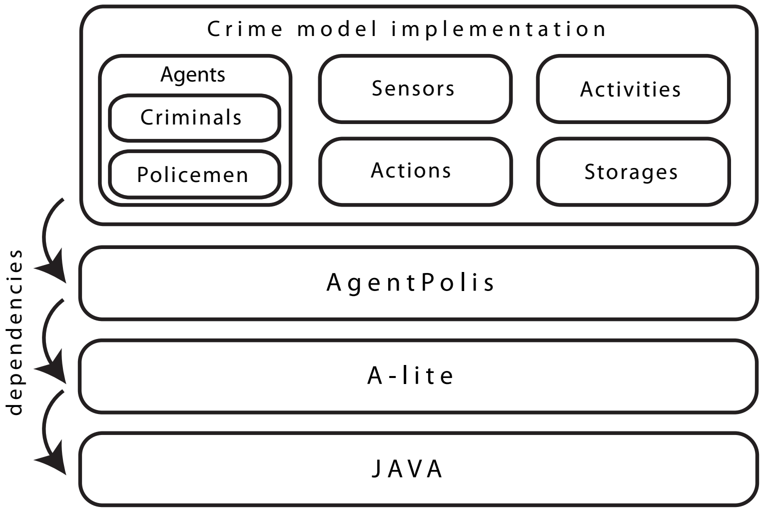


Figure 4.1: Scheme of dependencies - Java, A-lite, AgentPolis, crime model implementation

As it was repeatedly mentioned above, this project is focused on agent-based crime modelling. Thus, the application runs agent-based simulation. As a core for the simulation A-lite platform was used. AgentPolis, which is also built on A-lite platform, was used and extended to model criminals and policeman and their behaviour.

4.1 Programming language

Java SE 1.6[1] is the programming language used to program the main application. There are many reasons for using Java. Java is object oriented programming language. Agent-based simulation is, as the name suggests, based on the agents, small independent units. Objects in Java architecture are quite similar. Thus, Java, as object-oriented programming language, is suitable for the development of agent-based simulation applications. In addition to that there are some further advantages of Java language, e.g. it is platform independent, it uses garbage collector, there are many existing libraries which can be re-used and among others syntax is quite simple which makes the code well readable and understandable.

What's more, A-lite and especially AgentPolis are written in Java. And for the reason that the source code of A-lite and AgentPolis was available for use in this project, it was possible to modify whatever we needed.

4.2 A-lite

A-lite[15] is a multi-agent general platform developed by Agent Technology Center[2] at Faculty of Electrical Engineering[3] of Czech Technical University in Prague[4]. It is focused on large-scale agent-based simulations.

A-lite provides support for implementation when building any multi-agent systems. As the authors note "The goals of the toolkit are to provide highly modular, variable, and open set of functionalities defined by clear and simple API. The toolkit does not serve as a pre-designed framework for one complex purpose, it rather associates number of highly refined functional elements, which can be variably combined and extended into a wide spectrum of possible systems." [15]

A-lite does not provides us with particular implementations of an environment, agents, their behaviour or pre-defined interactions. It provides us tools, classes and methods on which it is possible to build particular simulation and A-lite also provides the basic visualisation engine.

As it was noted before, A-lite is programmed in Java as well.

4.2.1 Important features and architecture

A-lite is event-based, it means that one of the main components is event queue (`EventQueue` class). All the events (any action in simulation) which are supposed to be processed are added to `EventQueue` which is implemented as a priority queue. For this is the very core of an simulation, it is desired to keep `EventQueue` as simple as possible to ensure the high efficiency. Every event (`Event` class) which should be processed in the simulation is added to `EventQueue`. When an event is added to `EventQueue`, its priority is set according to the time when the event is supposed to be processed. The simulation time runs and when a start-time

[1] Object-oriented programming language, http://www.oracle.com/technetwork/java/javase/
[2] Agent Technology Center website - http://agents.felk.cvut.cz/
[3] CTU - Faculty of Eletrical Engineering website - http://www.fel.cvut.cz/
[4] Czech Technical University website - http://www.cvut.cz/

of an event comes up the event is passed to event handler (`EventHandler` interface) which process the event. Each event has its duration defined. When the time of duration elapses, object that added the event to event queue is notified. Those actions are managed by event processor (`EventProcessor`) which is the core class of whole A-lite platform.

All information about outer world are stored in storage (`Storage` class). However these `Storage` classes aren't directly accessible. When any agent wants to find something out about the outer world (environment) it is necessary to use so-called sensor (`Sensor` class). It provides agent with the possibility to observe the environment. Of course, agents can access only such information which are supposed to be accessed by ordinary agents. If agent wants to perform an action which is going to change the content of any storage he uses `Action class`. `Action` classes are very similar to `Sensor` classes. The main difference is in the purpose. These classes are divided to keep the right sense of the programming code. `Sensors` are designed to directly access information from storages, while `Actions` do not usually provide any information, `Actions` are supposed to change the content of storages.

It should be noted that agents which we wrote about in Section 4.2 are not agents as we would supposed. `Entities` in A-lite, as those mentioned agents are called, are just really simple patterns to be used for future design of ordinary agents.

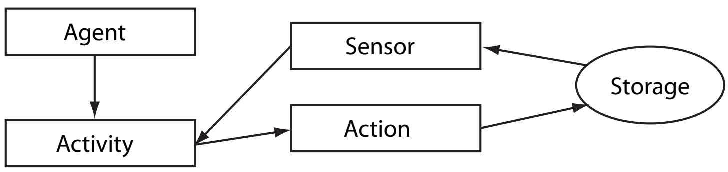


Figure 4.2: Scheme of interaction between agent and environment

4.3 AgentPolis

AgentPolis is an agent-based simulation framework for simulation of movement of people and vehicles in urban environment. The same way as A-lite, Agentpolis is developed by Agent Technology Center at CTU FEE.

AgentPolis is built on A-lite platform. It uses support which is provided by A-lite - tools, classes, methods etc. for agent-based simulation. AgentPolis implements particular agents, it designs an environment with all its parameters and it defines how the agents behave or how the interactions among them work.

4.3.1 Urban environment

As this agent-based simulation framework focuses mainly on the simulation of movement it is important to use good representation of individual positions and roads which connect positions together. In AgentPolis, positions and roads are represented by nodes and edges in a graph. Each node represents one position and roads are represented by edges.

4.3.2 Sensor, Action classes and Sensing interface

`Sensors` and `Actions` have already been described in Section 4.2.1. To recall it, sensors can be specified as read interfaces for interactions between `Activities` (respectivelly Agents, if you like) and `Storages`. `Actions` can be specified as write interfaces. Agents cannot access storages directly. However, it is important to note `Sensing` interface. Sensing interface makes it possible to announce to `Agent` that some other object interacts with him. Particular sensing is registered (usually in the storage) and when other entity wants to interact with agent who registered the sensing, the entity gets sensing reference from storage and it calls appropriate method to announce the interaction to interaction partner. The method has to be implemented by agent who registered the sensing. It should be mentioned that agents actually do not interact with each other or do not register or call sensings. `Activities` do that.

4.3.3 Agents

`Agent` class in AgentPolis extends `Entity` class from A-lite. It provides agents with ability to access other important classes as `Storage` (through `Sensor` or `Action`), `EventProcessor`, `Environment` and, last but not least, to perform activities (`Activity` class).

There are two types of `Agents` in AgentPolis. First is usual Agent with type `Adult`. `Adult` represents the most people. Adults have a job, they usually have a car or bicycle, they can entertain, eat out, do shopping etc. Of course, they can use public transport as well. Then there is another type of Agents called `Eternal Worker`. It is a special type of `Agent` whose purpose is to work as a public transport driver. `Eternal Worker` does not have any usual needs as desire for entertainment, sleeping etc. He does not do anything else than driving public transport mean for the whole duration of simulation.

4.3.4 Activity class

`Activity` class is completely new feature which wasn't provided by A-lite. It was created in AgentPolis because `Agent` class can't perform any action or do anything as itself. It is required to create an instance of any class extending `Activity` class (abstract class). For instance, when `Agent` wants to move from node A to node B, he can't just perform action like `agent.moveFromAtoB()` but it is necessary to create an instance of `Activity` class which provides `Agent` with the possibility to move from node A to node B. In fact, all the actions are performed by instances of `Activity` class because instances of `Agent` classes represent above all the agents as an object with states, not as an agent with a set of all behavioural rules and possible actions. One of the reasons for that is to keep the modularity of AgentPolis on high level because with the `Activity` architecture as it is now, you can completely change the behaviour of any `Agent` and you do not have to change any logical structure of `Agent` class.

4.3.4.1 Life cycle

`LifeCycleActivity` is main `Activity` which encapsulates whole life of an agent. LifeCycleActivity is also only `Activity` which is directly referenced from Agent instance. When

Agent is initialized his `LifeCycleActivity` is created and the reference is saved in Agent's local variable. `LifeCycleActivity` is then started by calling `born()` method.

Usual Agent, called Adult in AgentPolis, use `LifeCycleActivity` to repeatedly start `DayCycle` (see Section 4.3.4.2). On contrary, Eternal Worker agent does not have any `DayCycle`. `EternalWorker`'s `LifeCycleActivity` (class `EternalWorkerLifeCycleActivity`) directly creates an instance of `Activity` (which has to implement `WorkingActivity` interface) that never ends. `WorkingActivity` of `EternalWorker` makes him works forever in the same way.

Both, `Adult` and `EternalWorker` are instances of `Agent` class. However, when Agent is initialized, different `LifeCycleActivity` classes are assigned depending on agent type. If the type of newly created agent is Adult then `AdultLifeCycleActivity` is assigned as agent's `LifeCycleActivity`. And respectively if the type of new agent is Eternal Worker then `EternalWorkerLifeCycleActivity` is assigned. This design helps to keep high level of modularity which is crucial for agent-based simulation frameworks.

4.3.4.2 Day cycle

DayCycle Activity represents an ordinary day of Adult agent. It determines particular activities he does during day. Agent goes through many states and according to the previous state the new one is processed. Adult has a job, he can entertain, go shopping, eat out or sleep at home. This activity starts individual activities according to the current state of an agent — as activities representing working in office, spending time in the shops or entertainment centers or sleeping. As it was written above, this Activity is used only by agents of type Adult.

4.3.5 Movement and transport

Movement in AgentPolis is implemented in the way that agents can be either at the exact position or somewhere on the edge. All the position information are stored in EntityPositionStorage. Imagine an example where an agent stands at a position and he wants to reach neighbour node which the current node is directly connected with. It does not depend which mean will be used. GoingToActivity is created and then method `goTo()` is called. Duration of movement is calculated and then `Event` is added in `EventQueue` with the priority set to duration time added to the current time of simulation. The event is supposed to notify `GoingToActivity` when the agent reaches the desired node. When agent reaches new position, his position within `EntityPositionStorage` is changed to new node.

There are more possibilities how an agent can move. The principle is always the same as it was described above. However, an agent can use different means for transport. Some part of agents own a car or bicycle which can be used to accelerate their transport. In addition to that, there is public transport existing in AgentPolis. If people do not use their own car or bicycle they can use metro, bus or tram.

When an agent needs to get from one node to another and those nodes are not directly connected intelligent planning is used. AgentPolis generally uses A-star algorithm to plan the route.

4.3.6 Visualization

AgentPolis uses visualization engine from A-lite. `VisManager` (class from A-lite) manages visualization and it works on the basis of layers. You can create and register a layer and then add some visualization objects as points, lines etc. (with a possibility to change color and size settings) Visualization process usually acquires necessary information from various storages. See Figure 4.3 for the screen shot of visualization.

4.3.7 Statistics

Statistics are narrowly related to visualization. However, it does not originate from A-lite. Similarly to visualization it does not have any influence on the simulation. There are two parts into which the statistics can be divided.

Statistics in visualization Statistics in visualization are only the extension of visualization. It provides some important and interesting statistics about the simulation run and behaviour of agents, transports etc.

StatisticsStorage Storage from which Statistics in visualization mainly gets information. Whenever it is needed, some important records can be added into `StatisticsStorage`. A content of StatisticsStorage does not have any influence on the simulation. Its only purpose is to keep information which are important to be remembered from the run of simulation and next purpose is to provide information to all the objects working with the stats — mostly Statistics in visualization.

4.4 Crime model implementation

Mainly AgentPolis was used for an implementation of the model which was proposed in Chapter 3. Actually, the code of our project is just relatively heavily changed AgentPolis framework. According to our proposed model there were many changes made on AgentPolis. We disabled all Adult agents, all Eternal Worker agents, all cars and bicycles and all public transport means. However, many changes were made and lots of new features were added. The most of new features and changes are obvious regarding Chapter 3. Main design elements are similar to AgentPolis. In this section, we will describe how particular changes were implemented in our project. See Figure 4.3 for a screen shot of the visualization of our implementation.

4.4.1 Important notes

Randomness All random choices (unless otherwise stated) are generated from a uniform distribution.

Random decision If a probability $P \in \langle 0, 1\rangle$ is given for some random event, then a random number p from interval $\langle 0, 1)$ is generated and it will be decided to allow the event in case that $p < P$.

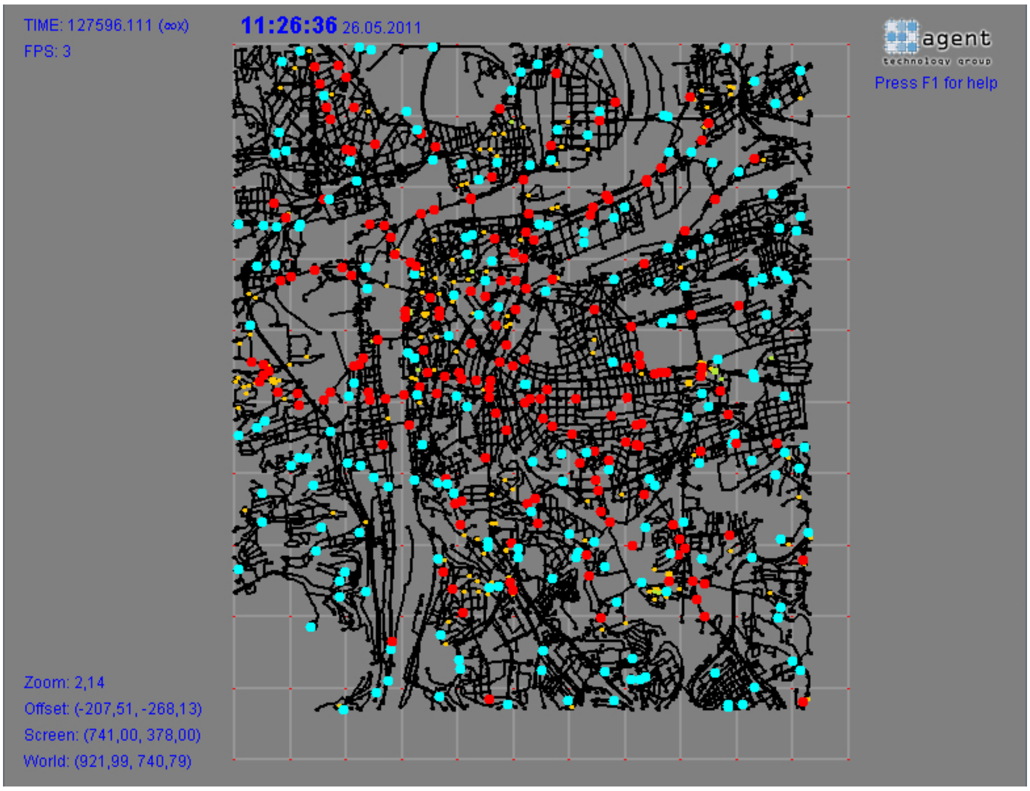


Figure 4.3: Crime model implementation screenshot

4.4.2 Environment

There were no changes that would worth noticing in comparison to AgentPolis or A-lite.

4.4.3 Storages

As we mentioned in Chapter 3, storages are mainly used for keeping information about outer world. However, design of storages varies a lot. Only storages which were designed or changed for the purpose of this project will be described in detail.

4.4.3.1 CrimeStorage

The main purpose of `CrimeStorage` is to store information about the crimes that are just being committed. It is stored in `HashMap<String, List<SensingCrime»` object. Where `String` key is the identification number of `Node` and value is `List<SensingCrime>`. When a criminal registers that he is committing the crime, it means that reference to him is added to `List<SensingCrime>` related to current `Node` where the criminal started the offence. When the crime is finished, the reference is deleted from the list. In addition to that, there is a `HashMap<String, Activity>` collecting all `WorkingAsPoliceman` and `CommittingCrime` activities. When an agent starts the eternal `WorkingActivity` he registers this activity in `CrimeStorage`. However, it is used only for the purposes of statistics.

4.4.3.2 PoliceStorage

`PoliceStorage` is used mainly as a central control component of police forces, although it does not keep the sense of storages to store information about the outer world. Shared memory keeping experiences with crime commitment attempts is stored in `PoliceStorage`. In addition to that, it keeps the list of policeman who are currently on duty. Although currently this feature does not have almost any sense, it could be quite useful in the future if one would want to allow the private life of policemen. Then, `PoliceStorage` keeps information about the desired distribution of police forces and factual current distribution of policemen. Information is kept in integer arrays.

`PoliceStorage` extends `EventBasedStorage` class. It means that storage can perform some action by itself at a given time. When `PoliceStorage` is initialized an event is added in `EventQueue`. When the `Event` is processed in `EventProcessor` it calls `handleEvent(Event)` method. `handleEvent` method adds a new event into `EventQueue`. Time interval between individual events is 10 minutes of simulation time. Besides, `handleEvent` gathers information about current police forces distribution and current state and content of the memory and it refreshes the array which determines the desired distribution of police forces (principles of refresh process are described in Section 3.4.2.2).

4.4.3.3 SectorStorage

The crucial part of `SectorStorage` is `HashMap<String, List<Node»` map. It stores List of Nodes which are related to given `Sector`. Sector is determined by its identification number. The map is constructed along with the environment. Each `Node` which is added into Environment is paired with the sector number. Find a sector which `Node` is related to is quite easy if you know the coordinates because you can calculate the borders of every single sector. But it would be difficult to find all Nodes which lies within given sector. This is the reason why map was created.

4.4.4 Sensors and Actions

There were a few new `Sensors` and `Actions` added as a part of our project. However, those classes do not provide us with any special feature which would worth noticing.

4.4.5 Agents

As it is not desired to model the day cycle of criminals and policemen with its usual parts (sleeping, working, entertainment, shopping) Eternal Worker design was used as a template for our new agent types. Two new agent types were created — `EternalCriminal` and `EternalPoliceman`. Their design is very similar to Eternal Worker. Both, criminal and policeman are represented by instances of Agent class. And because agent types are set to `EternalCriminal`, resp. `EternalPoliceman`, agent's `LifeCycleActivity` is set to

EternalCriminalLifeCycleActivity, resp. EternalPolicemanLifeCycleActivity. Both noted LifeCycleActivities mainly encapsulate never-ending WorkingActivity — EternalCriminalLifeCycleActivity encapsulates CommittingCrime and EternalPolicemanLifeCycleActivity encapsulates WorkingAsPoliceman. Both mentioned WorkingActivities represent never-ending loop. The loop goes through cycle of states. The cycle represents usual sequence of activities of ordinary criminal, resp. policeman.

4.4.5.1 WorkingActivity activity

The most important part of WorkingActivity is nextState() method. Not generally but it is true for the WorkingActivities of criminals and policemen. It works on the basis of finite-state machine. Each time when this method is called some actions are performed according to the current state and then Agent(respectively it is Activity because the states are changed inside CommittingCrime activity) changes the current state to the new one. The change is processed according to set of if-then rules in the body of nextState() method.

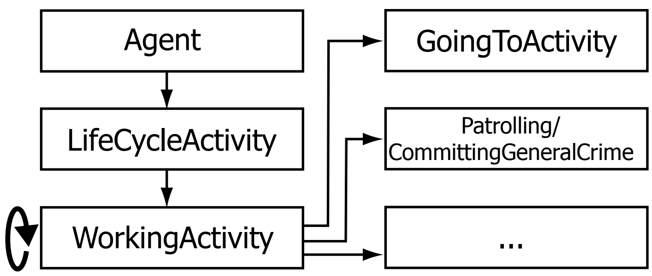


Figure 4.4: Activity scheme of criminals and policemen

4.4.6 Criminal

In this section CommittingCrime Activity will be mainly described. Criminal is using several Sensors for reading the environment variables — TimeSensor (access to EventProcessor and current time value), PositionSensor (access to information about current position), CrimeSensor (access to the records about crimes), SectorSensor (converting Node identification numbers to Sector numbers and vice-versa). Several parameters which influence behaviour of a criminal are set according to the simulation settings.

4.4.6.1 Committing crime

Committing particular crime is represented by separated Activity called CommittingGeneralCrime (abstract class). For the purpose of this project where no victims are interacted CommittingNullVictimCrime class was created. The act of committing a crime is started by a method with two parameters determining minimal and maximal duration. Particular duration of the crime is randomly chosen value from interval $\langle minDuration, maxDuration \rangle$. At the beginning of commitment, agent uses CrimeSensor to register Sensing related to himself

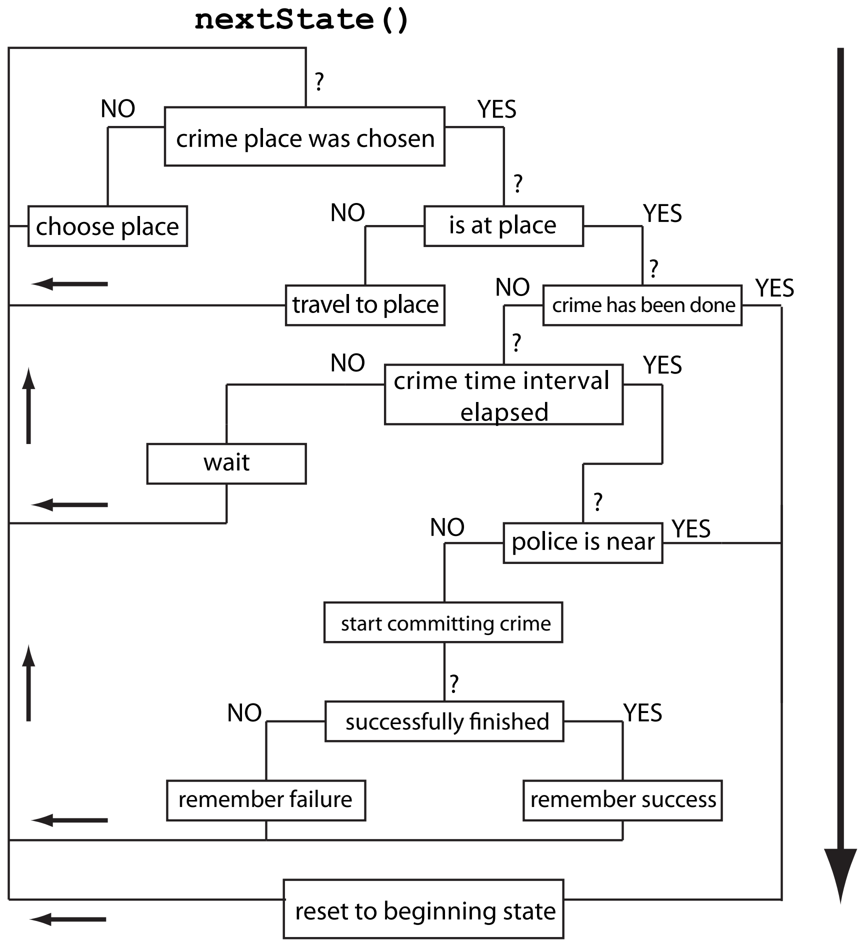


Figure 4.5: Scheme of decisions in nextState() method

in `CrimeStorage`. This Sensing is registered because if a policeman wants to interrupt this very crime event, he needs some mean to announce the interruption to the criminal.

After the `Sensing` registration `CommittingGeneralCrime` activity uses `TimeSensor` to add a new event to `EventQueue`. The event notifies the activity that the duration of commitment elapsed. If a criminal had been interrupted before the criminal was notified about the completed commitment, the notification is ignored. Otherwise, Sensing is removed from `CrimeStorage` (to remove the information about crime commitment at the node which is used by policemen for crime detection).

In the case of interrupted commitment, a policeman calls `crimeFailed()` method which is implemented by `SensingCrime` interface. Policeman acquires the reference to SensingCrime object from `CrimeStorage`. `crimeFailed()` method also removes the Sensing reference from `CrimeStorage` and it changes the state of activity to prevent potential notification from `TimeSensor`. `CommittingGeneralCrime` activity is immediately ended and the results of this activity are processed.

4.4.6.2 Processing crime commitment results

When `CommittingGeneralCrime` activity is ended, parent `Activity` is notified and has access to information whether the child `Activity` finished successfully (crime success) or whether the child Activity failed (crime interrupted). `CommittingCrime` is parent Activity of `CommittingGeneralCrime` in this case. When `CommittingCrime` activity is notified that `CommittingGeneralCrime` activity was ended, `CommittingCrime` activity records the experience. If the experience is positive or negative, it depends on the success of crime commitment. After this, criminal changes to its initial state and whole crime sequence starts from beginning.

4.4.6.3 Checking of the presence of police

The method that checks if no policeman is near is used in the `CommittingCrime` activity. There is a probability that the criminal fails to check, however the probability is initially set to be 0. If criminal does not fail he looks around for a policeman. Searching for policemen is implemented as breadth-first search on the graph structure that represents roads and positions. See Figure 4.6 for illustration. Red numbers in the picture describes the searching order of the edges. The search starts at current node and it cut off the nodes which are further than given distance (initially 50 meters). Algorithm iterates through the nodes chosen by breadth-first-search algorithm (more info about BFS in [12]) and checks if there is any policeman at the node — PositionSensor is used for that purpose. If police-awareness radius is set to 0 then only current node is checked.

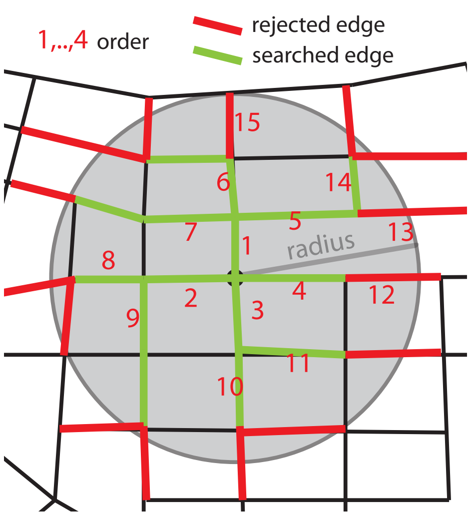


Figure 4.6: Breadth-first search illustration

4.4.7 Policeman

Implementation of policeman is very similar to the implementation of criminal. As in the previous case, `WorkingAsPoliceman` activity will be mainly described because this activity represents the never-ending working loop of a policemen. A finite-state machine is again in `nextState()` method. When the initial state is processed and a policeman is supposed to start patrolling, an instance of `Patrolling` activity is created and then the activity is started. One of important constructor parameters is the duration of patrolling activity (process of Patrolling was described in detail in Section 3.4.2.4).

4.4.7.1 Checking for a crime commitment around

CrimeSensor is used to check whether any crime is being committed around the current node of a policeman. Breadth-first search is used in CrimeSensor to check this. Search is constrained again by maximal distance, measured from current node (set according to simulation configuration, initially 120 meters). For each node which is examined, CrimeSector is accessed and CrimeSensor checks if there is any `SensingCrime` object registered in the storage. If any Sensing was registered in relation to the current node then it is added in List of all SensingCrime objects in the radius. Finally, one is randomly chosen and the reference is returned to the policeman(actually to Patrolling activity).

4.4.7.2 Processing patrol results

If no `SensingCrime` objects were found then null is returned and Patrolling continues until either, some commitment is discovered or the patrolling is finished because the patrolling duration elapsed. If an object which is instance of `SensingCrime` interface was returned from `CrimeSensor` then `Patrolling` activity is finished and marked as a successfully ended. Next, policeman has to move to the node where the crime is being committed. If the crime is still being committed when policeman arrives then crimeFailed() method is called from `WorkingAsPoliceman` activity. `crimeFailed()` method is implemented by `SensingCrime` interface and its function is described above in Section 4.4.6.1. If criminal who was spotted by a policeman has managed to leave before the policeman arrives then the procedure is the same as in the previous case. However, no `crimeFailed()` method is called. If the crime was discovered by a policeman before the duration of patrol elapsed, the time needed for finishing the patrol is calculated and new `Patrolling` activity is started with shortened duration at the same sector. Duration is given by newly calculated time. When policeman finishes his patrol he changes to his initial state and whole patrolling sequence starts from beginning.

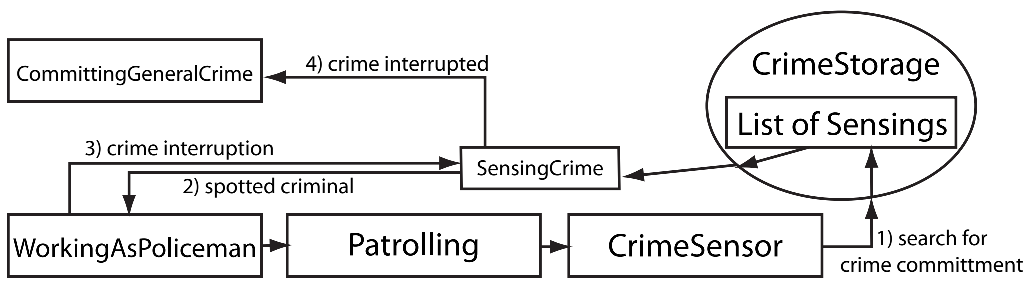


Figure 4.7: Scheme of crime interruption

4.4.8 Memory

To represent the memory with all the experiences, new class was created. It is called `MultiArmedBandit` as it some methods are partially based on Multi-armed bandit algorithms. There are two main variables stored in `MultiArmedBandit` class. The first is `Queue<Experience>` where a new experience are recorded. The second important variable is an array of doubles that stores evaluation of "expected rewards" from each single sector.

4.4.8.1 Sector choice algorithm

Always when the memory is refreshed (described in Section 4.4.3.2), all experiences which are older than given expiration time are removed from the memory. Then new array of "expected rewards" for each single sector is calculated. In addition, there is one more array. Indices are stored in this array, according to how the values of rewards array are sorted. Indices array is then used for roulette choice. First 10 percent of highest expected rewards are taken and according to their value, the roulette choice is used to pick the one. The sector which is represented by the chosen element is returned.

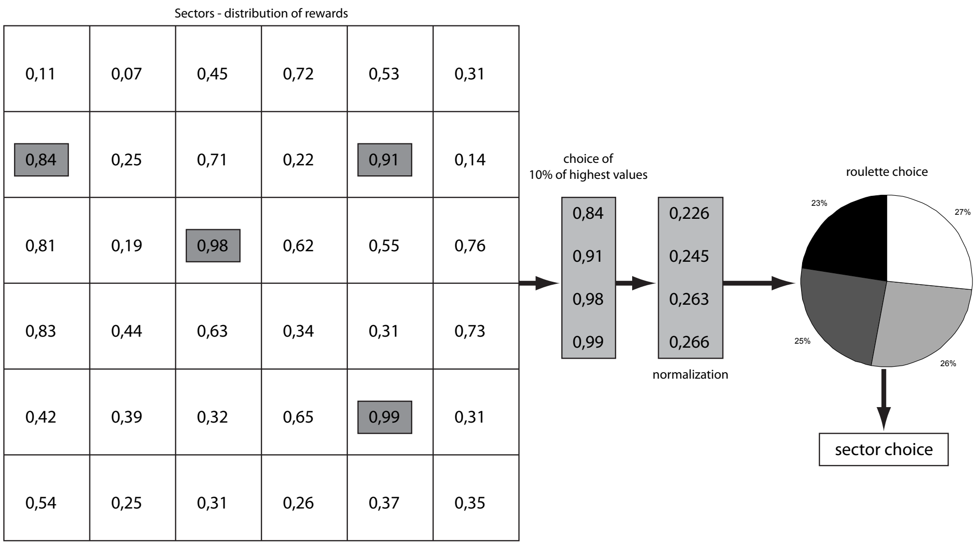


Figure 4.8: Sector choice algorithm

4.4.9 Evaluation of results

We have used MATLAB[5] as a basic auxiliary and evaluation tool. MATLAB was used for the batch file preparations, preprocessing of log files after the simulation and finally, as a main visualization and evaluation tool.

[5]MATLAB is a high-level language and interactive environment for technical computing. – http://www.mathworks.com/products/matlab/

As a part of our crime model implementation, logging utility was also included. We have created logging utility which writes the records into the log file. With this logging utility, it is possible to record almost any message. As some simulations were very memory and computationally demanding we recorded only the basic events. Moreover, for final experimentation, we disabled the visualization and reporting of AgentPolis platform to keep the memory size in the acceptable interval.

List of events that we record by specially created logging tool:

- Criminal is born
- Policeman is born
- Criminal started committing crime
- Crime was successfully committed
- Crime commitment was interrupted
- Policeman finished his patrol without any crime spotted
- Policeman spotted the crime but the criminal had already left

Whole experimentation process involves many steps. Now we will describe single steps. Experimentation progresses this way:

Batch file preparation As described in Section 5.1.4 each experiment took many repeated runs of the simulation, particularly from about 50 to 1500 iterations. We started the simulation 5 times for each scenario setting. It would be demanding to manage the executions of simulation manually. Therefore, we used MATLAB to create batch file that repeatedly started simulation with different setting.

Simulation The real time duration of one simulation run differs a lot. From about 30 seconds to 35 minutes. For the whole time of simulation, our logging tool was recording all the important events. As stated above, simulation itself runs on Java Virtual Machine as a compiled JAR file.

Preprocessing As there are usually many events recorded during the simulation, enormous amount of data is generated (usually from about 700 MB to 10 GB). Thus it is necessary to preprocess log files and get only a few summary files (usually not more than about 50 MB). MATLAB environment is used for preprocessing.

Evaluation and visualization Finally, files which we generated in the previous step are processed again. This time, we only evaluate success rate and sum the absolute numbers of events according to the subject of examination. MATLAB is also used for this step. At the end we visualize experiments data in the charts and observe the results.

4.5 Conclusion

As our implementation of urban crime uses A-lite and AgentPolis which are relatively well coded, code is clear and it is easy to fully understand or modify it. Therefore, the implementation of our model was not unreasonably complicated. A-lite simulation platform is also optimized for computational performance so the computational time is very acceptable especially when only ordinary Adult agents are simulated. In comparison to Adult agents, which are really simple and do not use any special deciding and interaction technique, criminals and policemen have higher computational requirements. More about performance of our implementation is written in next chapter in Section 5.4.

Chapter 5

Experiments

In previous chapters, we have proposed a model and described how we implemented it in our project. We performed a few experiments in order to verify the correctness of our implementation, to examine the model and the effects of changes of scenario parameters and to analyse key properties of the implemented model.

Experiments are divided in two parts. First, experiments performed on one single sector (let's call it "single-sector" experiments). And second, experiments performed on the whole city (let's call it "multi-sector" experiments). More information about these single-sector and multi-sector experiments is written in following section.

5.1 Experiment setting

5.1.1 Space configuration

5.1.1.1 City

As we mentioned in Section 3.3 and Section 4.3.1 the city where agents live consists of nodes and edges. Therefore, simulation needs a map of nodes and edges which represents the city. The city we have modelled is very similar to Prague. We used source maps from OpenStreetMap project[1] and then we edited the source maps in JOSM[2]. As the city of Prague is quite large, we cropped the selection. Upper left corner and bottom right corner of the selection were: the first, latitude 50.110991°N, longitude 14.3790116°E and the second, latitude 50.0440144°N, longitude 14.494981°E[3]. See Figure 5.1 for the illustration of the city.

5.1.1.2 Single-sector city

It isn't always necessary to use complete city map for the experiments. There are cases when one sector is enough. For example, when we perform an experiment where we compare various counts of criminals and policemen. Only difference between using one sector or using

[1] OpenStreetMap website - http://www.openstreetmap.org/
[2] Java OpenStreetMap editor - http://josm.openstreetmap.de/
[3] Coordinates are set according to Geographic coordinate system

Figure 5.1: Map that was used for multi-sector experiments

more sectors is that when you use only one sector the results are more likely to include deviation errors. Except of that, the results should be the same. Of course, you would have to modify your results according to the area size.

How was the single-sector city created? We analyzed whole city map which we used for multi-sector experiments and we got a few statistics which were used for creation of one sector which would represent average sector in the city. See the statistics in Table 5.1. A few things should be mentioned. sectors that are situated at the edge of the city are not involved in the statistics because they are often somehow influenced by their position, the edges with no node within were removed and finally top 10 percent and bottom 10 percent with highest and lowest counts of nodes were removed. It is reminded that the size of sector is initially 800×800 meters.

Finally, we artificially created a sector where we uniformly spread the nodes in a network where the nodes were connected with each other in columns and in rows. Sector was created according to the measured statistics.

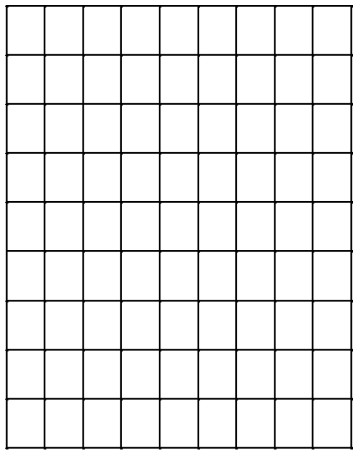


Figure 5.2: Map that was used for single-sector experiments

Stats	**Value**
Total number of sectors	110
Total number of nodes	8271
After removal of some sectors	
Total number of sectors	58
Total number of nodes	5096
Average number of nodes in sector	87.86

Table 5.1: Nodes and edges statistics

5.1.2 Scenario parameters

A few parameters were chosen as scenario parameters. The scenario parameters were changed many times regarding to the aim of our experiment. Other parameters stay the same without the possibility to be changed.

List of scenario parameters:

- Number of criminals
- Number of policemen
- Minimal crime duration
- Maximal crime duration
- Police range of sight
- Police-awareness of criminals(distance)
- Police movement speed
- Memory expiration time — determining how long the experiences will be valid

5.1.3 Evaluation metrics

During the experiments which were performed on one single sector, we focused mainly on the crime success rate. We usually measured the crime success rate for each setting of scenario and evaluated it. Crime success rate is calculated as a number of successfully finished crimes divided by number of attempts to commit a crime. In multi-sector experiments, we also focused on the crime success rate. However, we examined the progress of absolute number of crimes as well. We usually use the value which says how many measured events have happened in the last 12 hours.

5.1.4 Simulation setting

All our experiments last 35 days of simulation time. However, as we wanted to avoid the deviation errors and we wanted to experiment on the simulation that would be stable without any unpredictable influences (caused e.g. by massive immediate actions of agents at the beginning) we removed the first 5 days from experiment records. Thus, we used records which had been recorded on days in interval $\langle 6, 35 \rangle$. Every experiment was repeated five times because it was desired to use averaged results in order to avoid any deviation.

5.2 Single-sector experiments

5.2.1 Dependency on number of policemen, number of criminals and crime duration

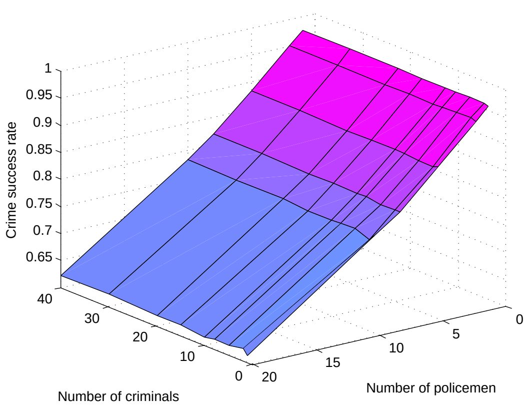
This experiment was focused on the crime success rate and its dependency on the number of policemen, the number of criminals and crime duration. We were changing parameters that determines number of criminals and number of policemen. We studied the results for four different settings of crime duration intervals. Finally, we compared and discussed the results. For particular settings of simulation parameters, see Table 5.2.

Parameter	**Value**
Number of criminals	1,2,5,8,10,15,20,30,40
Police-awareness(dist.)	120 m
Number of policemen	1,2,5,8,10,20
Police-awareness(probab.)	100%
Crime duration	(1-3),(4-6),(9-11),(19-21)mins
Police movement speed	4.7 kmph
Police range of sight	120 m
Warm-up events	5 per agent
Memory expiration time	5 days

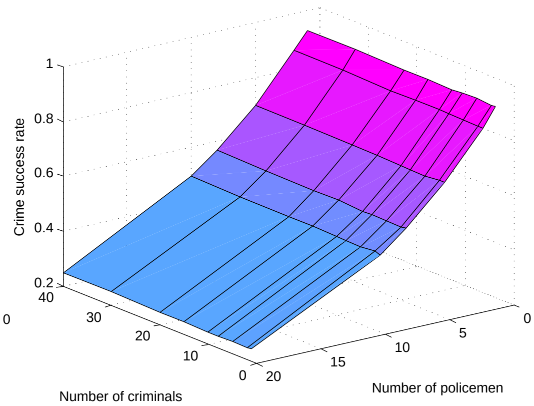
Table 5.2: Scenario parameters - Dependency on number of policemen, number of criminals and crime duration

The results of the experiment can be seen in Figures 5.3, 5.4 and 5.5. Figure 5.5a and Figure 5.5b were created by averaging the results along the number of criminals, resp. the number of policemen axis.

We can observe a few very interesting things in the results. First, the influence of the number of criminals on the success rate is surprisingly low. See Figure 5.5a. You can see that success rate is almost constant, regardless of the number of criminals. In spite of that, you can still see that the success rate falls a little when the crime duration is 19–21mins. Difference between the success rates of the case with 1 criminal and of the case with 40 criminals is -4,5%. It could be caused by the fact that policemen interrupts still the same number of crimes but criminals do not commit as many crimes as before. Thus, the crime success rate is decreasing. On the contrary to dependency on the count of criminals, there is strong dependence on the count of policemen, as it is seen in Figure 5.5b.

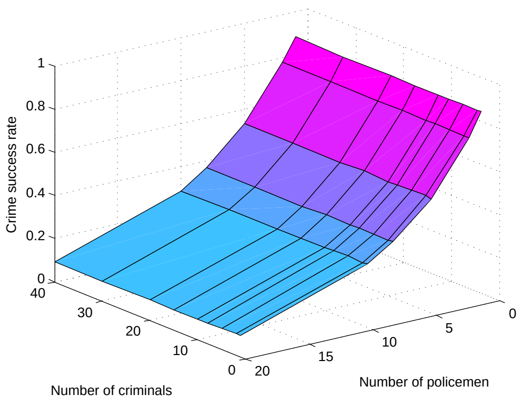


(a) Crime duration 1–3 minutes

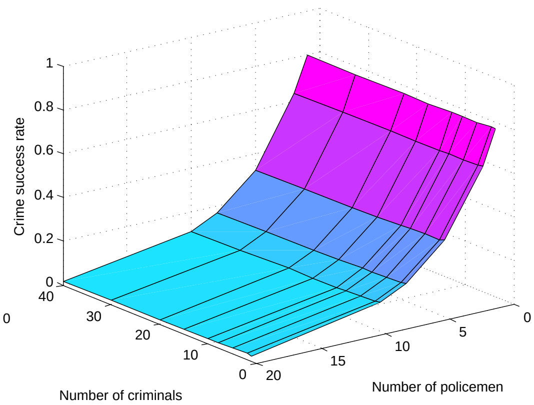


(b) Crime duration 4–6 minutes

Figure 5.3: Crime success rate - Dependency on the numbers of criminals and policemen

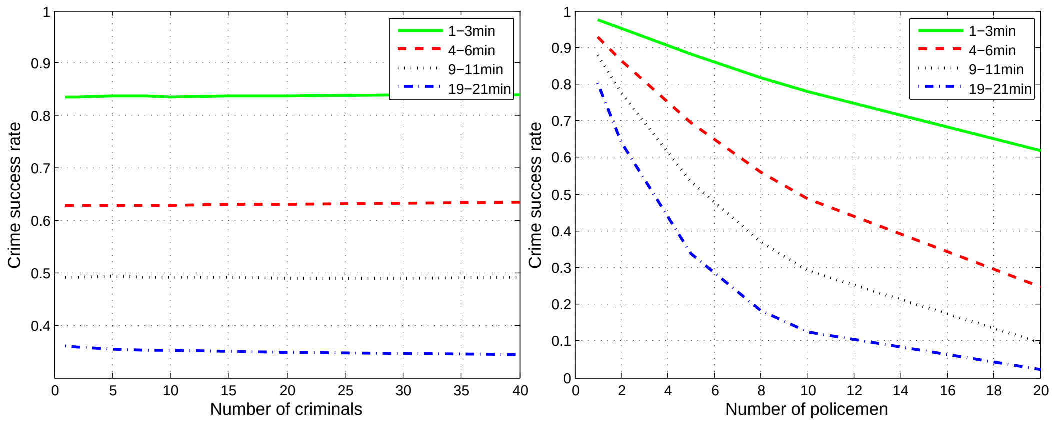


(a) Crime duration 9–11 minutes



(b) Crime duration 19–21 minutes

Figure 5.4: Crime success rate - Dependency on the numbers of criminals and policemen



(a) Success rate - dependency on the number of criminals

(b) Success rate - dependency on the number of policemen

Figure 5.5: Success rate - dependency on the number of agents

Second, you can notice the effect of crime duration interval on the success rate. It is noticeable especially in Figure 5.5b. It is obvious that when the crime duration is shorter, it is much harder to interrupt the commitment — the chance that the criminal would be spotted during the commitment is lower. On the opposite, when a criminal spends a long time by committing a crime then the probability of his offense interruption increases. When there is only one policeman in the sector, the difference is not so obvious but with increasing number of policemen, the difference in success rate becomes well observable.

And the third interesting thing is that when the crime duration is shorter then the success rate is linearly decreasing along with the presence of more policemen. However, when the crime duration is longer, the curve representing the fall of success rate becomes to look similarly to $\sqrt{x}$ function.

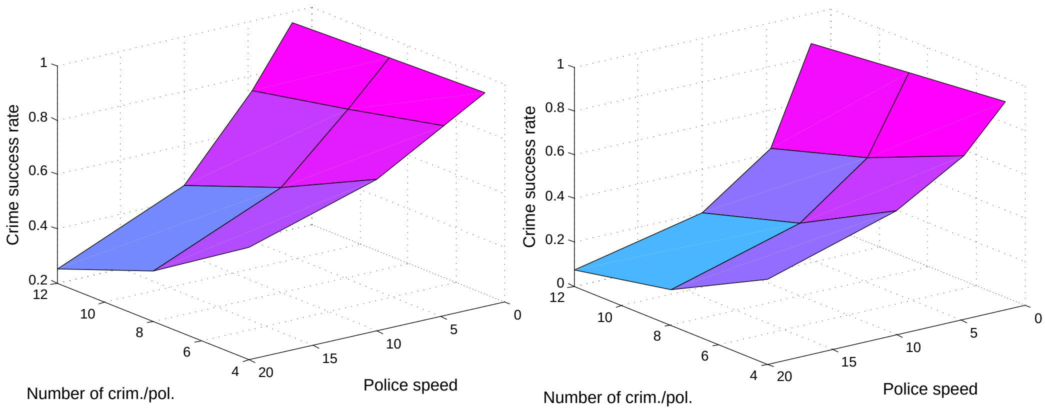
5.2.2 Dependency on number of agents, police speed and crime duration

This experiment was noted in Section 2.3 as one of possible simulation scenarios. Imagine what would happen if all policemen would ride a bike instead of going on foot. Would it change the crime success rate? See Table 5.3 for the scenario parameters which were used for this experiment. Four different speeds were used in the simulations — 1.5, 4.7, 10.0, 20.0 kmph. Number of criminals, number of policemen and crime duration were also changed a few times in the simulation to observe some secondary effects of these changes.

First, it is necessary to explain properly the details of simulation. In contrast to previous experiment the simulation didn't run with all combinations of agents' count. It means that during simulations 3 different agents' count were used. 4 criminals and 4 policemen, then 8 criminals and 8 policemen and finally 12 criminals and 12 policemen. However, as we know from previous experiment, resulting crime success rate does not depend much on criminals count. Therefore, we can speak mainly about the number of policemen. The results of the experiment can be seen in Figure 5.6 and 5.7.

Parameter	Value
Number of criminals	4,8,12
Police-awareness(dist.)	120 m
Number of policemen	4,8,12
Police-awareness(probab.)	100%
Crime duration	(1-3),(4-6),(9-11)mins
Police movement speed	1.5, 4.7, 10.0, 20.0 kmph
Police range of sight	120 m
Warm-up events	5 per agent
Memory expiration time	5 days

Table 5.3: Scenario parameters - Dependency on number of agents, police speed and crime duration

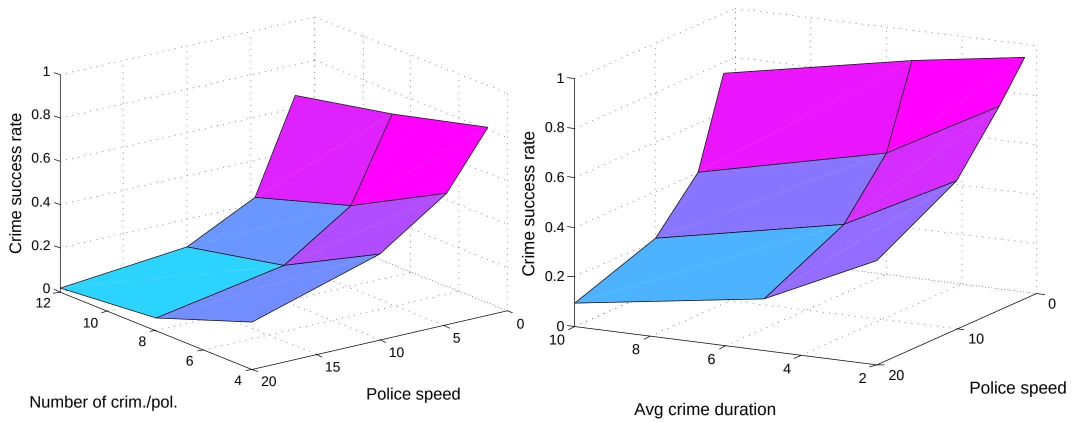


(a) Crime duration 1–3 minutes (b) Crime duration 4–6 minutes

Figure 5.6: Crime success rate - Dependency on police speed and the number of agents

In Figures 5.6a, 5.6b and 5.7a, we can see 3 plots, each for one single crime duration. In Figure 5.8, the results from previous 3 pictures were averaged along the axis which determined the number of criminals (resp. policemen). This time the results correspond to the expectations. As we know from the experiment in Section 5.2.1, as the number of policemen grows the success rate falls. This is quite obvious in the plots. Plots for three different crime duration intervals are very similar. We can observe again the same thing as in the previous experiment — with longer crime duration, even small change has big effect on the crime success rate. On the contrary, when the crime duration is short, the effect of speed change is not so obvious and the success rate does not fall so steeply.

Finally, in Figure 5.7b we can see that the success rate is really decreasing along with the increase in police speed. This observation verified the general assumption that it really depends on the speed of police forces. If the police forces were faster they would be able to catch more criminals. Police patrols would be able to cover larger areas and the police actions would be more effective because they would catch more criminals after the announcement of



(a) Crime duration 9-11 minutes

(b) Dependency between police speed and crime duration

Figure 5.7: Crime success rate - Dependency on police speed and the number of agents

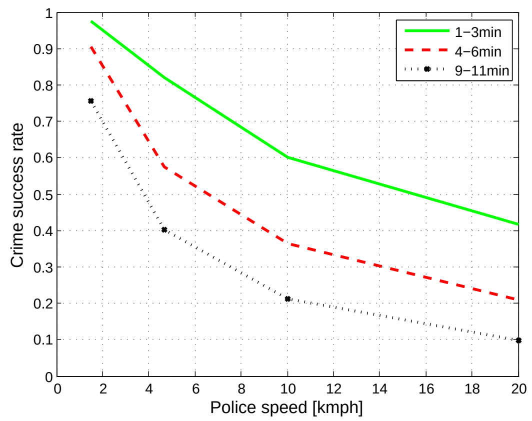


Figure 5.8: Dependence between police speed and success rate

the crime.

5.2.3 Dependency on number of agents, police range of sight and crime duration

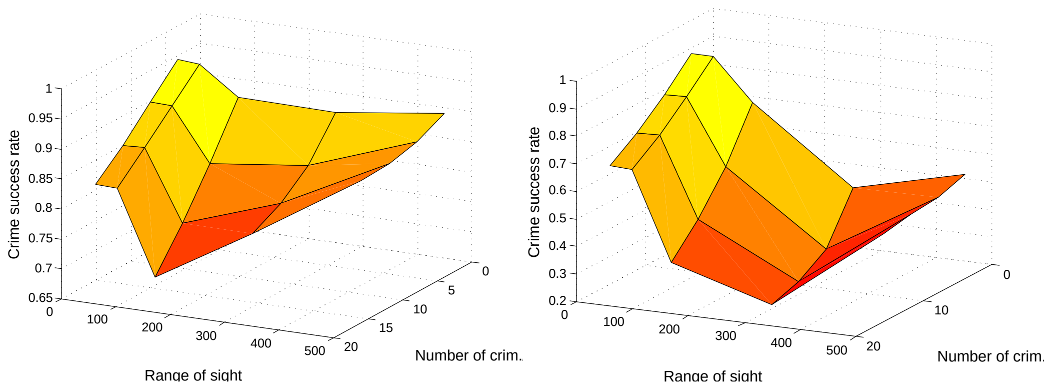
Interesting parameter for observation is the range of sight of police forces. Imagine the case when the police forces cannot see further than a few meters. This wouldn't be good because they would have to patrol and search for the crime only randomly. On the contrary, if they could see to the long distances it would be good because no criminal could hide. See Table 5.4 for scenario parameters.

This experiment was focused on the changes of the range of sight of police forces. It means that we were changing parameters of the behaviour which was described in Section 4.4.7.1.

Parameter	**Value**
Number of criminals	4,8,12,16
Police-awareness(dist.)	120 m
Number of policemen	4,8,12,16
Police-awareness(probab.)	100%
Crime duration	(1-3),(4-6),(9-11)mins
Police movement speed	4.7 kmph
Police range of sight	10, 50, 120, 300, 500 m
Warm-up events	5 per agent
Memory expiration time	5 days

Table 5.4: Scenario parameters - Dependency on number of agents, police range of sight and crime duration

When you have a look at the results you can see quite interesting figures which do not exactly correspond with our expectations.

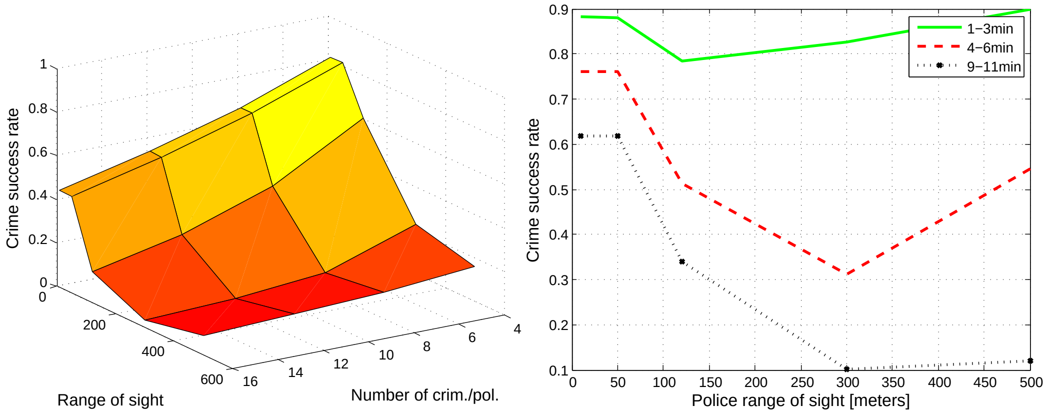


(a) Crime duration 1–3 minutes (b) Crime duration 4–6 minutes

Figure 5.9: Crime success rate - Dependency on the police range of sight and number of agents

How could we explain the break in figures? We can observe that there is a break in figures — in a parallel way with the axis determining the number of agents, crossing the axis determining Range of sight at about 300 meters. Our explanation is that policeman needs a minimal range of sight to be able to patrol effectively. This is clear — we can see in Figure 5.10b that the minimal range of sight should be about 150 meters. However when policeman can see much further (more than 300 meters) and he spots the crime, he fixes this place in his memory and he travels to this place. And it is obvious that when the crime lasts short time, it is difficult to catch the criminal still at the crime commitment place. In short, policemen are attracted by distant crimes instead of patrolling the surrounding where some crime could occur soon.

To make a conclusion, it would be surely good to have unlimited range of sight. However



(a) Crime duration 9-11 minutes

(b) Success rate - Averaged dependency on range of sight

Figure 5.10: Crime success rate - Dependency on the police range of sight and number of agents

it is necessary to centrally control the policeman or provide them with a good communication channel in order to send the closest one to the crime place and use the police forces effectively. This could be good proof that the communication of police forces and good scheduling of police activities are really important.

5.2.4 Dependency on criminals' awareness distance, crime duration and the number of agents

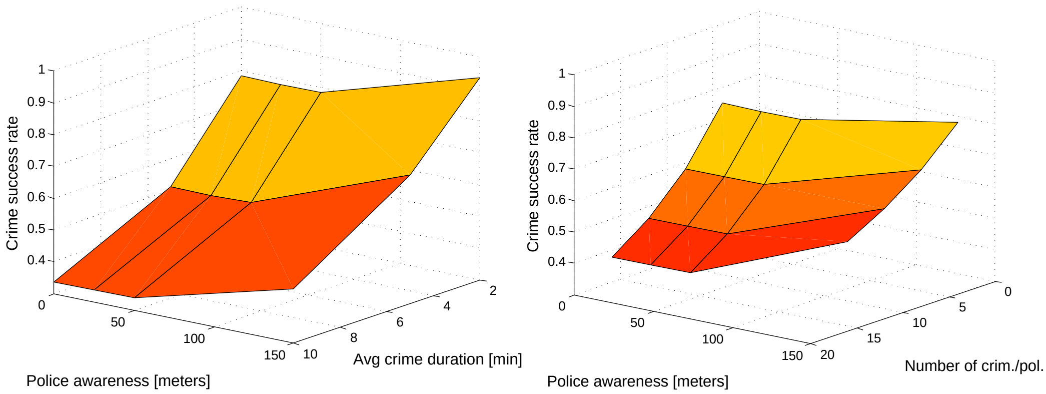
Experiment with criminals' awareness distance is similar to the experiment in Section 5.2.3. This time, we adjust the range of sight of criminals determining the radius in which they check whether no police forces are around when they want to commit a crime. We can find the example of this experiment even in the real world. There are often the situations when you are in a hurry, you need to catch your appointment and you drive a little faster than it is allowed by the law. This time you are not careful and there can be also police officers somewhere far and watch you through their binoculars. However, when you arrive at the crossing and you see the police officer, you will definitely slow down. See, this is the influence of awareness distance.

To remind the use of awareness distance, always when the criminal wants to start committing a crime, he checks whether there is any policeman around. Awareness distance determines the radius which the criminal searches for the policeman in. See Table 5.5 for scenario parameters of this experiment. The results are not surprising. In Figure 5.11b, we can see that success rate increases when the criminals are more aware of police officers. However, it is obvious in Figure 5.12a that although the crime success rate is higher, the total number of crime attempts steeply falls.

To get a conclusion, from these results we can see one important thing. It is good when the police forces are visible and can prevent the crimes but also it is important to keep

Parameter	Value
Number of criminals	4,8,12,16
Police-awareness(dist.)	0,25,50,150 m
Number of policemen	4,8,12,16
Police-awareness(probab.)	100%
Crime duration	(1-3),(4-6),(9-11)mins
Police movement speed	4.7 kmph
Police range of sight	120 m
Warm-up events	5 per agent
Memory expiration time	5 days

Table 5.5: Scenario parameters - Dependency on criminals' awareness distance, crime duration and the number of agents



(a) Success rate - dependency on awareness and crime duration

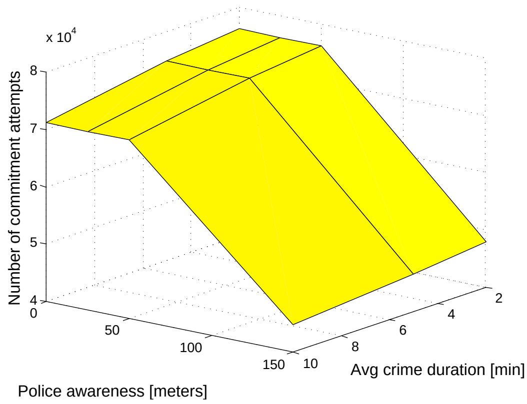
(b) Success rate - dependency on awareness and the number of agents

Figure 5.11: Crime success rate - Dependency on criminals' awareness distance, crime duration and the number of agents

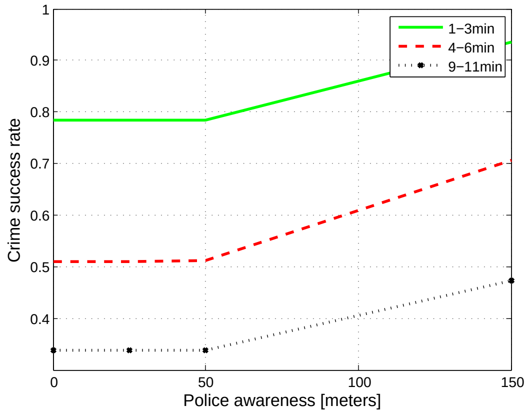
in mind that then the police forces cannot evaluate their activities according to the crime success rate. There are two possibilities. Police will be visible — they will prevent many crimes but they will not catch almost any criminal. Or police will not be visible — police will arrest a lot of criminals but also the total number of crimes would be much higher.

5.3 Multi-sector experiments

Despite the fact, we examined most basic relations in Section 5.2, we performed a few multi-sector experiments as well. The purpose of multi-sector experiments is not to observe basic dependencies and influences of different scenario parameters. In the case of multi-sector experiments, we rather examine the direct interactions between policemen and criminals and the influence of their memory.



(a) Absolute number of commitment attempts



(b) Crime success rate - Dependency on awareness

Figure 5.12: Dependency on criminals' awareness distance, crime duration and the number of agents

5.3.1 Criminals and policemen interactions

In this section, we closely examined only one single sector. We examined the absolute numbers of events which had happened in the sector during whole simulation run and then analyzed it with regard to the time. See Table 5.6 for scenario parameters. We remind that all following experiments were performed in the multi-sector city.

Parameter	**Value**
Number of criminals	300
Police-awareness(dist.)	50 m
Number of policemen	300
Police-awareness(probab.)	100%
Crime duration	(9–11)mins
Police movement speed	4.7 kmph
Police range of sight	120 m
Warm-up events	5 per agent
Memory expiration time	1, 5, 15 days
Memory ε constant	0.05, 0.30, 0.80

Table 5.6: Scenario parameters - Multi-sector experiments - Interaction between criminals and policemen

5.3.1.1 Average memory and average exploitation/exploration ratio

See Table 5.7 for particular setting of this experiment. There are two important figures which should be noted. In Figure 5.13, you can see the numbers of different crime events

Parameter	Value
Memory expiration time	5 days
Memory ε constant	0.30

Table 5.7: Particular scenario setting for the observation of criminals-policemen interactions

which happened in the selected sector. In Figure 5.14, you can see the same events but this time in detail. The results are obviously good because as we can see in figures criminals and policemen interact just in the way as we expected at the beginning and as we planned. First, the wave of criminals comes and it is still growing. Then, the number of successfully committed crimes falls and the number of interrupted crimes grows. Therefore, the total number of commitment attempts also falls. Next, you can see in Figure 5.14, that a few policemen still patrol, however after a while everybody leaves the sector — criminals were not successful, and then policemen did not find any criminals during their patrol.

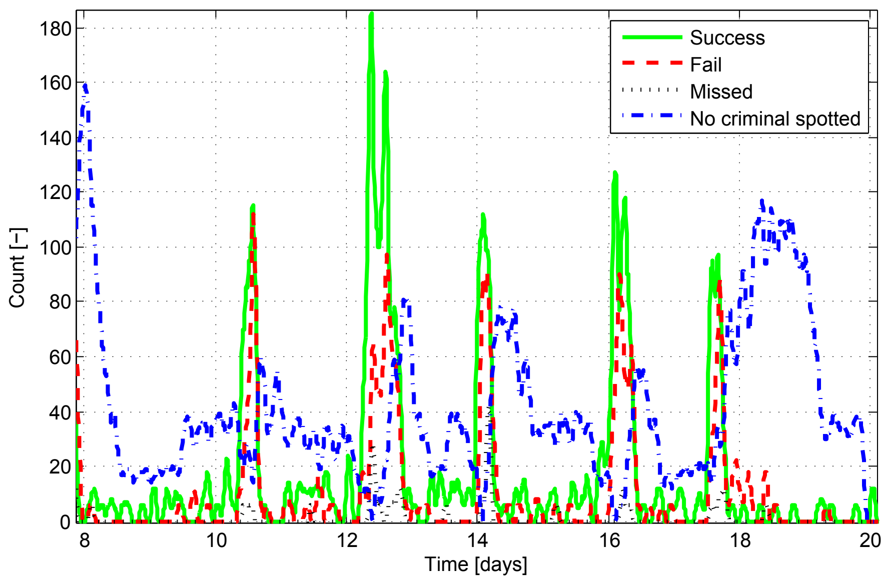


Figure 5.13: Numbers of crime events in the selected sector

5.3.1.2 Short memory and a lot of exploration

In this particular scenario (for parameters see Table 5.8) we examine the case when agents prefer mainly the random exploration and they remember only short interval of time.

In Figure 5.15, we can see that the interactions are not so obvious in the contrary to the previous case. It's caused mainly by the fact that agents behave randomly in most cases and their memory quickly develops.

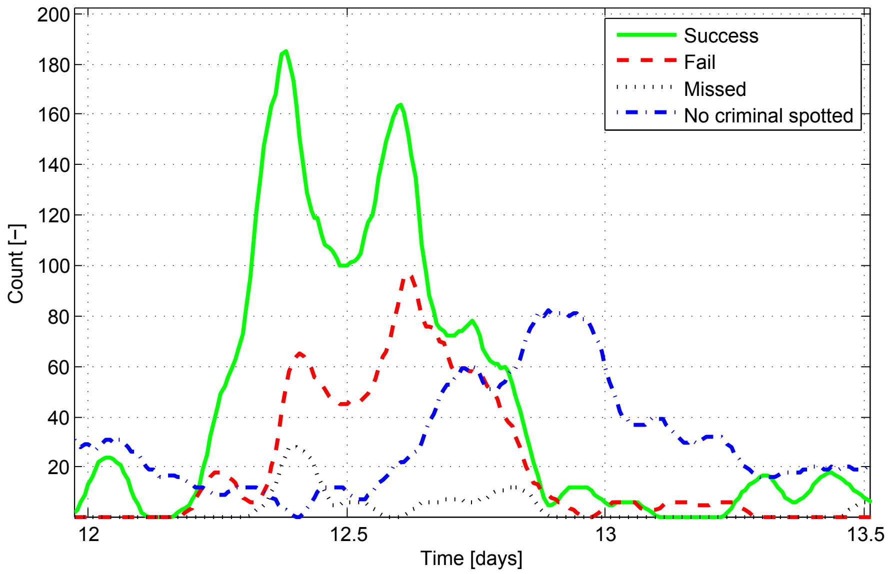


Figure 5.14: Numbers of crime events in the selected sector - in detail

Parameter	**Value**
Memory expiration time	1 day
Memory ε constant	0.8

Table 5.8: Particular scenario setting for the observation of criminals-policemen interactions - short memory

5.3.1.3 Long memory and a lot of exploitation

This experiment is almost the exact opposite to the previous one. For exact scenario parameters see Table 5.9.

Parameter	**Value**
Memory expiration time	15 day
Memory ε constant	0.05

Table 5.9: Particular scenario setting for the observation of criminals-policemen interactions - long memory

In opposite to the previous experiment, we can see in Figure 5.16 that the speed of changes in memory is very slow. And in addition, agents in most cases choose to pick the most attractive sector. From this reason, total numbers of events are even bigger than the numbers in Figure 5.13.

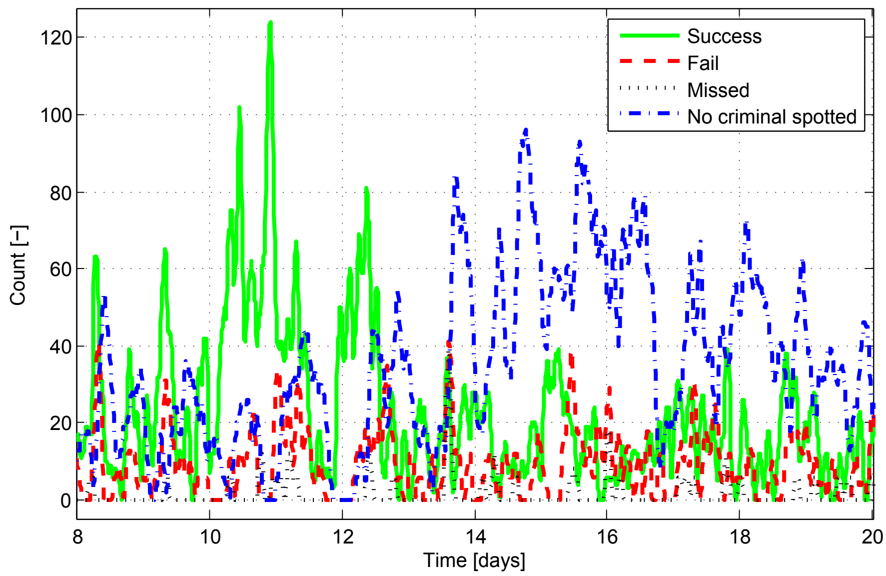


Figure 5.15: Numbers of crime events in the selected sector - short memory

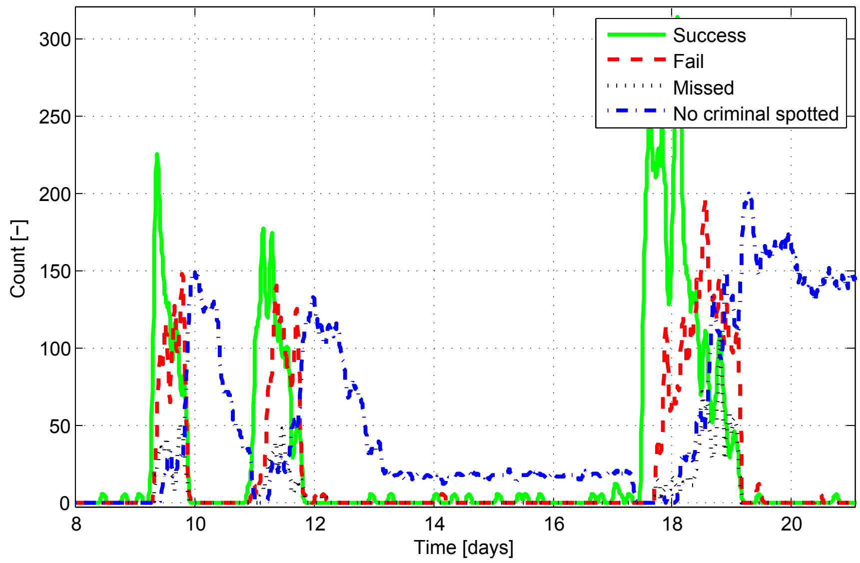


Figure 5.16: Numbers of crime events in the selected sector - long memory

5.3.2 Crime success rates - dependency on the memory

Finally, in Table 5.10 and in Figure 5.17 you can see the crime success rates calculated from previously addressed events. Differences in the crime success rate are not anyhow significant.

There is only about 7 percent difference. So we can hardly make any presumptions.

	1 day	5 days	15 days
0.05	0.6284	0.6033	0.5803
0.3	0.6402	0.6299	0.6267
0.8	0.6471	0.6473	0.6491

Table 5.10: Crime success rate - Dependency on memory setting (epsilon constant – memory expiration time)

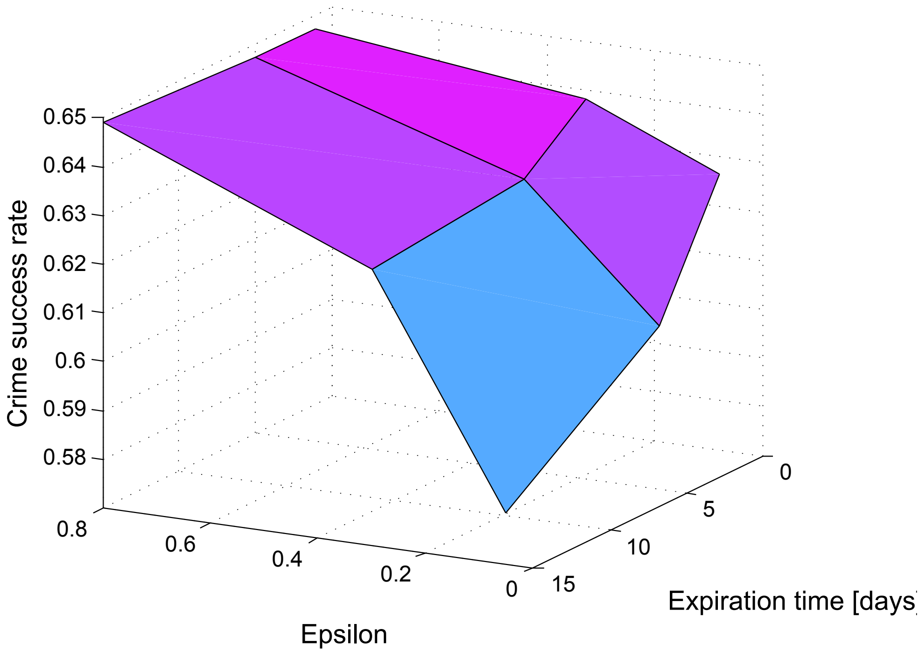


Figure 5.17: Crime success rate - Dependency on memory setting (epsilon constant – memory expiration time)

5.4 Computational performance

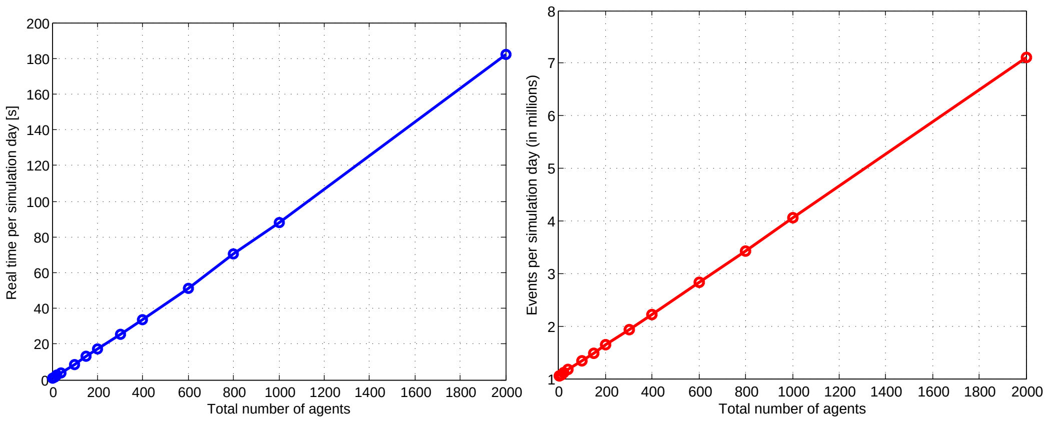
First, it is important to mention that most experiments were performed on the computer with hardware as it is listed in Table 5.11. Therefore, most performance statistics have to be examined considering the hardware.

In Figure 5.18, we can observe two different plots describing the performance of our implementation in dependence on the number of agents. We should also note that AgentPolis simulation containing only ordinary Adult agents without any criminals and policemen can simulate on similar performance level the environment containing about 10000–50000

Processor	Intel Core 2 Duo, E8400, 3,00Ghz
RAM	4 GB
Operating system	Windows 7, 32-bit

Table 5.11: Hardware configuration of the computer that was used for experimentation

agents. Measurement of implementation performance was performed on normal setting while changing the number of agents. We used multi-sector map for measurement simulation.



(a) Time dependence on the number of agents (b) Events count dependence on the number of agents

Figure 5.18: Computational performance

We experimented also with 5000 criminals and 5000 policemen in the environment. We observed that duration of simulation and the number of events still grow linearly.

5.5 Conclusion

It is obvious that in many cases agent-based crime simulation can be very useful. We have verified the correctness of our model and we observed many interesting things in our experiments. Of course, there is a lot of space for future experiments and for extension of scenario parameters. Although we haven't done any revolutionary finding, we demonstrated the possibilities which our model offers. Single conclusions regarding particular results were made after each experiment where it was important.

Chapter 6

Conclusions

We have learnt many interesting information about agents, agent-based modelling and especially agent-based crime modelling. We went through the related work. Examination of the related projects allowed us to understand many basic features of agent-based modelling (resp. crime modelling). Finally, essential part of those information was introduced in Chapter 2.

Thanks to knowledge which has been acquired, we proposed our own crime model. Proposed model was described in detail in Chapter 3. We proposed the fundamental parts of our model — especially environment and agents — with their behaviour, interactions and relations.

We got familiar with A-lite and AgentPolis agent-based simulation platforms. Our knowledge was then interpreted in Chapter 4 where we also described how we had implemented our own crime model.

When the crime model implementation was finished, we needed to verify its correctness. We created a few scenarios and ran the simulation. Finally, we analyzed the results and discussed their correctness and sense. As our results were quite similar to our expectations, they corresponded to the general idea of this system and all the results had a sense, we can state that we proposed good model. As it was noted before, we did not discover anything revolutionary in this work, however we showed that there is a promising future in agent-based crime modelling and that it is definitely worth of further research.

As A-lite platform and AgentPolis framework were used and fully understood, it is possible to extend this project in the future. And as it is built on the basis of A-lite and AgentPolis which are used by other people, this project can be used also by other people to further examine agent-based modelling of urban crime. Although, only agent-based crime modelling was mentioned, there are many very similar fields of interest which are almost the same and it would be possible to use this work only with minor adjustments. For instance, criminals and policemen can be substituted by fare evaders[1] and controllers.

6.1 Future work

A few examples of possible future works are given in this section.

[1] People who do not pay fares for public transports, trains etc

6.1.1 Victims

As we noted in Section 3.3.1, no victims are involved during crime events in this project. This could be quite interesting feature which could be examined further. There is a space to model an interaction between the criminal and his victim. And you can also change various victims and observe different results. By changing the victim types and the interaction, you can model different crime situations — pickpockets in public transports, burglars, car thief etc.

6.1.2 Agents, cars, companies

Although ordinary citizens, cars, companies etc. were omitted in this project, it would be interesting to observe the results if those entities were involved in the simulation. Criminals would react on the day time, on the count of citizens in the streets etc.

6.1.3 More different crime types

In addition to the project extensions which were proposed above, more different crime types could be modelled and implemented together. Then it would be interesting to observe the influence of a day time on the crime type which is committed in the most cases. For example, the results could be similar to the following: cars would be stolen mostly at night; during the peak of public transport, the pickpocketing would be the most committed crime type etc.

6.1.4 Simplicity

Although the proposed extensions are very interesting we can't forget one thing. It is important to keep some given level of simplicity. The reasons were mainly discussed in Chapter 3. It is good to have simulation which can simulate the behaviour of criminals with all extensions that were noted above. However, it is useless if you can't analyze the results correctly. And you probably wouldn't be able to analyze the results because the complexity of the model would be too high.

Bibliography

[1] BENENSON, I. – TORRENS, P. M. Geosimulation: object-based modeling of urban phenomena. *Computers, Environment and Urban Systems*. January 2004, 28, 1, s. 1–8.

[2] BONABEAU, E. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*. 2002, 99, Suppl 3, s. 7280–7287. doi: 10.1073/pnas.082080899. Dostupné z: <http://www.pnas.org/content/99/suppl.3/7280.short>.

[3] BORSHCHEV, A. – FILIPPOV, A. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. In *Proceedings of the 22nd International Conference*, Oxford, England, UK, 2004. System Dynamics Society.

[4] BOSSE, T. – GERRITSEN, C. 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*, AAMAS '08, s. 1129–1136, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-1-6.

[5] BOSSE, T. et al. LEADSTO: A Language and Environment for Analysis of Dynamics by SimulaTiOn. In ALI, M. – ESPOSITO, F. (Ed.) *Innovations in Applied Artificial Intelligence*, 3533 / *Lecture Notes in Computer Science*. Ottawa, Canada: Springer Berlin / Heidelberg, 2005. s. 193–203.

[6] BOSSE, T. et al. Agent-based vs. population-based simulation of displacement of crime: A comparative study. *Web Intelligence and Agent Systems*. 2011, 9, 2, s. 147–160.

[7] BRANTINGHAM, P. L. – J., B. P. Environment, Routine, and Situation: Toward a Pattern Theory of Crime. In CLARKE, R. V. – FELSON, M. (Ed.) *Routine activity and rational choice. Advances in criminological theory.*, 5. New Brunswick: New Brunswick, 1993. s. 259–294.

[8] COOK, P. J. The Demand and Supply of Criminal Opportunities. *Crime and Justice: An Annual Review of Research*. 1986, 7, s. 1–28.

[9] KELEMEN, J. *Umělá inteligence 3*, 1, Reaktivní agenti, s. 161–188. Academia, Praha, 2001.

[10] MACAL, C. M. – NORTH, M. J. Tutorial on agent-based modeling and simulation. In *Proceedings of the 37th conference on Winter simulation*, WSC '05, s. 2–15. Winter Simulation Conference, 2005. ISBN 0-7803-9519-0.

[11] MAHAJAN, A. – TENEKETZIS, D. Multi-Armed Bandit Problems. In HERO, A. O. et al. (Ed.) *Foundations and Applications of Sensor Management*. Springer US, 2008. s. 121–151.

[12] MAŘÍK, V. *Umělá inteligence 1*, 1, Řešení úloh a využívání znalostí, s. 33–66. Academia, Praha, 1993.

[13] MALLESON, N. – HEPPENSTALL, A. – SEE, L. Crime reduction through simulation: An agent-based model of burglary. *Computers, Environment and Urban Systems*. May 2010, 34, 3, s. 236–250. ISSN 01989715. doi: 10.1016/j.compenvurbsys.2009.10.005. Dostupné z: <http://dx.doi.org/10.1016/j.compenvurbsys.2009.10.005>.

[14] NETRVALOVÁ, A. *Úvod do problematiky multiagentních systémů* [online]. 2008. [cit. 9. 5. 2011]. Dostupné z: <http://www.kiv.zcu.cz/~netrvalo/phd/MAS.pdf>.

[15] CTU FEE, A. T. C. *A-lite* [online]. 2008. [cit. 13. 5. 2011]. Dostupné z: <http://agents.felk.cvut.cz/projects/>.

[16] PARUNAK, H. V. D. – SAVIT, R. – RIOLO, R. L. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In *Proceedings of the First International Workshop on Multi-Agent Systems and Agent-Based Simulation*, s. 10–25, London, UK, 1998. Springer-Verlag. ISBN 3-540-65476-3.

[17] SCHMIDT, B. Human Factors in Complex Systems: The Modelling of Human Behaviour. In MERKURYEV, Y. – ZOBEL, R. – KERCKHOFFS, E. (Ed.) *Proceedings of the 19th European Conference on Modelling and Simulation*, ECMS '05, s. 5–14. The European Council for Modelling and Simulation, 2005.

[18] SNOOK, B. Individual differences in distance travelled by serial burglars. *Journal of Investigative Psychology and Offender Profiling*. 2004, 1, 1, s. 53–66. ISSN 1544-4767. doi: 10.1002/jip.3. Dostupné z: <http://dx.doi.org/10.1002/jip.3>.

[19] VERMOREL, J. – MOHRI, M. Multi-armed Bandit Algorithms and Empirical Evaluation. In GAMA, J. a. et al. (Ed.) *Machine Learning: ECML 2005*, 3720 / *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005. s. 437–448.

[20] WRIGHT, R. T. – DECKER, S. H. – GEIS, G. *Burglars On The Job: Streetlife and Residential Break-ins*. Boston : Northeastern University Press, 1996. ISBN 1555532713.

Appendix A

List of Abbreviations

ABM Agent-based modelling

BFS Breadth-first search

GB Gigabyte

GHz GigaHertz

JAR Java Archive

MB Megabyte

RAM Random access memory

Appendix B

User guide

User guide can be also found on the CD. As there are not anything complicated on our CD there are only a few hints.

If you wish to compile the source codes on your own, follow these instructions:

- Set Main class to: Main - cz.agents.alite
- Use program arguments: cz.agents.urbansim.creator.OsmScenarioCreator -simdays 15 -crim 200,200 -crdur 9,11 -pros 120 -pspeed 4.7 -creps 0.25 -expt 432000000 -warmup 400 -awarpro 1.0 -awardist 50
- Use VM arguments: -Djava.util.logging.config.file=logging.properties -Dsun.java2d.d3d=false -Xmx1024M

If you wish to set your own scenario parameters, please, follow these descriptions:

- simdays - Length of the simulation in days
- crim - Number of criminals/policemen
- crdur - Minimal crime duration in minutes/Maximal crime duration in minutes
- pros - Police range of sight in meters
- pspeed - Police speed in kmph
- creps - Memory epsilon constant in interval <0.0-1.0>
- expt - Memory expiration time in millis
- warmup - Number of warm-up events in memory
- awarpro - Police awareness (probability)
- awardist - Police awareness (distance in meters)

If you wish to start only compiled application, use batch-files in /implementation/examples/ folder.

- start_big.bat - starts multi-sector scenario
- start_small.bat - starts single-sector scenario

Further instructions: Press F1 when running the application.

Appendix C

Content of the CD

documents/	- contains declaration and assignment
experiments/	- contains MATLAB scripts for generating the plots
implementation/	- contains our own implementation
examples/	- contains compiled file for running our implementation
start_big.bat	- batch file for multi-sector simulation execution
start_small.bat	- batch file for single-sector simulation execution
sourcecode/	- contains source code of our implementation
text/	- contains the text of our project
src/	- contains source files for LaTeX
Modry-thesis-2011.pdf	- text of this work
readme.txt	- readme file
users_guide.txt	- user's guide