



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# **Masterarbeit**

**Kim Beier**

**Simulation der Übertragung von tropischen  
Infektionskrankheiten zwischen Mensch und Moskito mit  
einem Multi-Agenten System**

*Fakultät Technik und Informatik  
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science  
Department of Computer Science*

Kim Beier

**Simulation der Übertragung von tropischen  
Infektionskrankheiten zwischen Mensch und Mosquito mit  
einem Multi-Agenten System**

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Stefan Sarstedt  
Zweitgutachter: Prof. Dr. Olaf Zukunft

Eingereicht am: 28. Juli 2015

**Kim Beier**

**Thema der Arbeit**

Simulation der Übertragung von tropischen Infektionskrankheiten zwischen Mensch und Moskito mit einem Multi-Agenten System

**Stichworte**

Multi-Agenten System, Epidemiologie, Dengue-Fieber, Malaria, West-Nile-Fieber, Chikungunya-Fieber, SIR-Modell, SEIR-Modell, Moskito, durch Vektor übertragen

**Kurzzusammenfassung**

In dieser Arbeit wird ein Multi-Agenten System realisiert, das die Übertragung zwischen Mensch und Moskito simuliert. Es können verschiedene tropische Infektionskrankheiten, deren Übertragung zwischen Mensch und Moskito erfolgt, simuliert werden. Außerdem ist es möglich eine Region, Quarantänestationen und Schädlingsbekämpfer zu definieren. Vor jedem Simulationsdurchlauf wird durch die Berechnung der Reproduktionsnummer ermittelt, ob es sich bei der Infektionskrankheit um eine Epidemie handelt. Die Auswertung durch eine SEIR-Epidemiekurve des Menschens erfolgt am Ende jeder Simulation.

**Kim Beier**

**Title of the paper**

Transmission simulation of tropical infectious diseases between human and mosquito with an multi-agent system

**Keywords**

Multi-Agent system , Epidemiology, Dengue-Fever, Malaria, West-Nile-Fever, Chikungunya-Fever, SIR-Model, SEIR-Model, Mosquito, Vector-borne

**Abstract**

In this paper, a multi-agent system is realized, which simulates the transmission of tropical infectious diseases between humans and mosquitos. The transfer of several tropical infectious diseases between humans and mosquitos can be simulated. It is also possible to define a region, quarantines and pest control. Prior to each simulation run it can be determined by the calculated reproduction number, whether the infectious disease is epidemic or not. At the end of each simulation an evaluation by an SEIR epidemic curve of humans is made.

# Danksagung

*An dieser Stelle möchte ich allen danken, die mich bei der Erstellung der Masterarbeit unterstützt haben.*

*Im Besonderen gilt mein Dank Ralf Rosen, der durch Korrekturen, Verbesserungsvorschläge und hilfreiche Diskussionen zur Qualität dieser Arbeit beigetragen hat.*

*Nicht zu vergessen sind an dieser Stelle meine Eltern, die mir das Studium mit ihrer fortwährenden Unterstützung überhaupt erst ermöglicht haben.*

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>viii</b>
<b>Abbildungsverzeichnis</b>	<b>ix</b>
<b>Listings</b>	<b>xi</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Ziel der Arbeit . . . . .	2
1.3 Aufbau der Arbeit . . . . .	2
<b>2 Grundlagen</b>	<b>3</b>
2.1 Begriffserklärung . . . . .	3
2.2 Vektor-übertragende Infektionskrankheiten . . . . .	4
2.2.1 Allgemein . . . . .	4
2.2.2 Dengue-Fieber . . . . .	4
2.2.3 Malaria . . . . .	5
2.2.4 Chikungunya-Fieber . . . . .	5
2.2.5 West-Nile-Fieber . . . . .	5
2.3 SIR-Modell . . . . .	6
2.4 Ausbreitungsmodelle . . . . .	7
2.4.1 Agentenbasiertes Modell . . . . .	7
2.4.2 Zellulärer Automat . . . . .	8
<b>3 Verwandte Arbeiten</b>	<b>10</b>
3.1 Multi-Agent-Systems in Epidemiology . . . . .	10
3.1.1 SEIR-Epidemiemodell . . . . .	10
3.1.2 Modell . . . . .	11
3.1.3 Komponentenzusammenspiel . . . . .	12
3.1.4 Funktionen . . . . .	13
3.2 Agent-Based-Model for the Spread of the Dengue Fever . . . . .	14
3.2.1 Modell . . . . .	14
3.3 An agent-based modeling for pandemic influenza in Egypt . . . . .	15
3.4 Vector-born Epidemie Modell . . . . .	15
3.4.1 Modell . . . . .	15

<b>4</b>	<b>Analyse</b>	<b>19</b>
4.1	Anforderungen . . . . .	19
4.1.1	Agent (Mensch) . . . . .	19
4.1.2	Moskito (Vektor) . . . . .	20
4.1.3	Umwelt . . . . .	20
4.1.4	Konfiguration . . . . .	21
4.1.5	Darstellung . . . . .	21
4.1.6	Technische Anforderungen . . . . .	22
<b>5</b>	<b>Entwurf</b>	<b>23</b>
5.1	Architektur . . . . .	23
5.1.1	Anwendungskern . . . . .	24
5.1.2	Fassade . . . . .	25
5.1.3	Benutzeroberflächen . . . . .	25
5.1.4	Adapter . . . . .	26
5.1.5	Dateisystem . . . . .	26
5.2	Laufzeitsicht . . . . .	27
<b>6</b>	<b>Realisierung</b>	<b>32</b>
6.1	Realisierungsumgebung . . . . .	32
6.2	Anwendungskern . . . . .	33
6.3	Fassade . . . . .	36
6.4	Benutzeroberflächen . . . . .	37
6.5	Adapter . . . . .	41
6.6	Dateisystem . . . . .	41
<b>7</b>	<b>Ergebnisse und Diskussion</b>	<b>42</b>
7.1	Simulationsumgebung . . . . .	42
7.2	Tropische Infektionskrankheiten - Parameter . . . . .	42
7.3	Ergebnisse . . . . .	46
7.3.1	Default . . . . .	46
7.3.2	Quarantänestationen . . . . .	49
7.3.3	Schädlingsbekämpfer . . . . .	56
7.3.4	Wiederinfizierung . . . . .	60
7.3.5	Quarantänestationen und Schädlingsbekämpfer . . . . .	63
7.4	Diskussion . . . . .	72
7.4.1	Dengue-Fieber . . . . .	72
7.4.2	Malaria . . . . .	72
7.4.3	West-Nile-Fieber . . . . .	73
7.4.4	Chikungunya-Fieber . . . . .	74
<b>8</b>	<b>Schluss</b>	<b>77</b>
8.1	Zusammenfassung . . . . .	77

8.2 Ausblick . . . . .	77
<b>Anhang</b>	
<b>A Realisierung</b>	<b>xii</b>
A.1 C# Code - InEnvironment() und InVirtualField() . . . . .	xii
A.2 XML-Struktur für XML-Datei . . . . .	xii
<b>B Ergebnisse und Diskussion</b>	<b>xiv</b>
B.1 Graph erstellen . . . . .	xiv
<b>C Inhalt der beigefügten CD</b>	<b>xv</b>
<b>Literaturverzeichnis</b>	<b>xvi</b>

# Tabellenverzeichnis

6.1	Realisierungsumgebung - Eigenschaften . . . . .	32
7.1	Simulationsumgebung - Eigenschaften . . . . .	42
7.2	Allgemein - Parameter . . . . .	42
7.3	Dengue-Fieber - Parameter . . . . .	43
7.4	Malaria - Parameter . . . . .	44
7.5	West-Nile-Fieber - Parameter . . . . .	44
7.6	Chikungunya-Fieber - Parameter . . . . .	45
7.7	Zusammenfassung der Ergebnisse nach Szenarien . . . . .	68
7.8	Zusammenfassung der Ergebnisse nach Krankheiten . . . . .	71

# Abbildungsverzeichnis

2.1	SIR-Modell (Entnommen aus: <a href="#">Khalil u. a. (2010)</a> ) . . . . .	6
2.2	Agentenbasiertes Modell (Entnommen aus: <a href="#">Macal und North (2010)</a> ) . . . . .	7
2.3	Agentenbasiertes Modell - Topologien (Entnommen aus: <a href="#">Macal und North (2010)</a> ) . . . . .	8
2.4	Zellulärer Automat (Entnommen aus: <a href="#">Sun u. a. (2012)</a> ) . . . . .	9
3.1	SEIR-Epidemiemodell (In Anlehnung an: <a href="#">Roche u. a. (2008)</a> ) . . . . .	10
3.2	Komponenten des Multi-Agenten-Systems (Entnommen aus: <a href="#">Roche u. a. (2008)</a> ) . . . . .	11
3.3	Komponenten des Agent-based Model (In Anlehnung an: <a href="#">Jacintho u. a. (2010)</a> ) . . . . .	14
3.4	Host-Vektor Modell (In Anlehnung an: <a href="#">Erickson u. a. (2010)</a> ) . . . . .	15
5.1	Architektur des Prototypen . . . . .	23
5.2	Ablaufdiagramm des Prototypen . . . . .	28
5.3	Ablaufdiagramm des Prototypen - Agent Process . . . . .	29
5.4	Ablaufdiagramm des Prototypen - Moskito Process . . . . .	31
6.1	Punkt in einem Polygon (In Anlehnung an: <a href="#">Orwant u. a. (2000)</a> ) . . . . .	34
6.2	Vor dem Start einer Simulation . . . . .	37
6.3	Nach den Einstellen der Daten für die Simulation . . . . .	38
6.4	Simulationsdurchlauf . . . . .	38
6.5	Virtuelles Feld eines Moskitos . . . . .	39
6.6	KonfigurationsTool im Initialzustand ohne gesetzte Parameter . . . . .	40
6.7	KonfigurationsTool nachdem Simulationsparameter eingestellt wurden . . . . .	41
7.1	Default - Dengue-Fieber . . . . .	46
7.2	Default - Malaria . . . . .	47
7.3	Default - West-Nile-Fieber . . . . .	47
7.4	Default - Chikungunya-Fieber . . . . .	48
7.5	Position der Quarantänestation - 1 Quarantänestation . . . . .	49
7.6	Positionen der Quarantänestationen - 2 Quarantänestationen . . . . .	49
7.7	Positionen der Quarantänestationen - 4 Quarantänestationen . . . . .	50
7.8	Positionen der Quarantänestationen - 8 Quarantänestationen . . . . .	51
7.9	Unterschiedliche Anzahl an Quarantänestationen - Dengue-Fieber . . . . .	52
7.10	Unterschiedliche Anzahl an Quarantänestationen - Malaria . . . . .	53
7.11	Unterschiedliche Anzahl an Quarantänestationen - West-Nile-Fieber . . . . .	54
7.12	Unterschiedliche Anzahl an Quarantänestationen - Chikungunya-Fieber . . . . .	55
7.13	Definition - Count Exterminator . . . . .	56

7.14	Unterschiedliche Anzahl an Schädlingsbekämpfern - Dengue-Fieber . . . . .	56
7.15	Unterschiedliche Anzahl an Schädlingsbekämpfern - Malaria . . . . .	57
7.16	Unterschiedliche Anzahl an Schädlingsbekämpfern - West-Nile-Fieber . . . . .	58
7.17	Unterschiedliche Anzahl an Schädlingsbekämpfern - Chikungunya-Fieber . . . . .	59
7.18	Definition - ReInfection . . . . .	60
7.19	Wiederinfizierung und keine Wiederinfizierung - Dengue-Fieber . . . . .	60
7.20	Wiederinfizierung und keine Wiederinfizierung - Malaria . . . . .	61
7.21	Wiederinfizierung und keine Wiederinfizierung - West-Nile-Fieber . . . . .	62
7.22	Wiederinfizierung und keine Wiederinfizierung - Chikungunya-Fieber . . . . .	62
7.23	Dengue-Fieber - 2 Quarantänestationen und 80 Schädlingsbekämpfern . . . . .	63
7.24	Malaria - 4 Quarantänestationen und 80 Schädlingsbekämpfern . . . . .	64
7.25	West-Nile-Fieber - 2/4 Quarantänestationen und 80 Schädlingsbekämpfern . . . . .	64
7.26	Chikungunya-Fieber - 4 Quarantänestationen und 40 Schädlingsbekämpfern . . . . .	65

# Listings

A.1	C# Code - InEnvironment() und InVirtualField() . . . . .	xii
A.2	XML-Struktur der XML-Datei . . . . .	xii
B.3	Matlab Code - Grapherstellung . . . . .	xiv

# 1 Einleitung

Dieses Kapitel gibt einen Einblick in die Thematik und erläutert das Ziel und den Aufbau.

## 1.1 Motivation

Weltweit erkranken jährlich unzählige Menschen in tropischen/subtropischen Ländern an tropischen Infektionskrankheiten, u.a. Dengue-Fieber, Malaria, West-Nile-Fieber, Chikungunya-Fieber. Überwiegend werden diese Erkrankungen durch Moskitos übertragen. 280 Personen erkrankten in Jamaica an Malaria Ende 2006 (06.11.2006 - 03.02.2007) ohne Todesfälle (Vgl. [World Health Organization \(2015e\)](#)). Im Oktober (15.10.2010) wurden der World Health Organization (WHO) 1500 laborbestätigte Infizierte und 15 Tote in Pakistan gemeldet (Vgl. [World Health Organization \(2015c\)](#)). Einen weiteren Ausbruch von Malaria gab es im Juli 2011 (07.2011 - 11.08.2011) in Europa. Am West-Nile-Fieber erkrankte wurden der WHO aus Albanien (2 Erkrankte), Griechenland, (22), Israel (6), Rumänien (1) und der russischen Föderation (11) gemeldet (Vgl. [World Health Organization \(2015g\)](#)). Zwei Jahre später, im Dezember 2012 (10.10.2012) gab es 18 bestätigte und 191 mutmaßlich an Dengue-Fieber Infizierte in Madeira, Portugal (Vgl. [World Health Organization \(2015d\)](#)). An Chikungunya-Fieber erkrankten im Dezember 2013 (06.12.2013) 2 Personen auf den Karibischen Inseln (französischer Teil)(Vgl. [World Health Organization \(2015b\)](#)). Ein weiterer Vorfall von Chikungunya-Fieber ereignete sich im Oktober 2014 (20.10.2014) in Montpellier, Frankreich. Dies war der erste Vorfall seit 2010, dass sich das Chikungunya-Fieber lokal auf 4 Personen ausbreitete (Vgl. [World Health Organization \(2015a\)](#)). Im November 2014 (28.11.2014) wurde der World Health Organization (WHO) die erste Entdeckung von West-Nile-Fieber in Brasilien gemeldet, die einen Menschen betraf (Vgl. [World Health Organization \(2015f\)](#)).

Diese Ausbrüche müssen überwacht werden, um mögliche Epidemien zu erkennen und einzudämmen.

## 1.2 Ziel der Arbeit

Das Ziel der Arbeit ist es ein Multi-Agenten System zu realisieren, dass die Übertragung von tropischen Krankheiten zwischen Mensch und Mosquito simuliert. Es soll möglich sein, Parameter zu verändern, um verschiedene tropische Infektionskrankheiten zu simulieren. Zudem soll es die Möglichkeit eine Region, Quarantänestationen und Schädlingsbekämpfer (Exterminator) zu definieren. Moskitos sollen nach dem SEI-Epidemiemodell und Menschen nach dem SEIR-Epidemiemodell den Krankheitsverlauf durchlaufen. Vor jeder Simulation wird die Reproduktionsnummer<sup>1</sup> ( $R_0$ ) aus den angegebenen Werten errechnet, sie soll angeben, ob es sich um eine Epidemie handelt. Die Auswertung der Simulation erfolgt am Ende, durch Darstellung einer SEIR-Epidemiekurve.

## 1.3 Aufbau der Arbeit

Die Arbeit ist in acht Kapitel gegliedert. Dieses Kapitel 1 bildet einen Einstieg in den Themenbereich, beschreibt die Motivation und das Ziel der Arbeit. In Kapitel 2 werden die notwendigen Grundlagen dieser Arbeit erläutert und umfassen eine Einleitung in die Begriffe der Epidemiologie. Des Weiteren werden in den Grundlagen die vektor-übertragenden Infektionskrankheiten, das SIR-Modell und die Ausbreitungsmodelle beschrieben. Im anschließenden Kapitel 3 werden die verwandten Arbeiten Multi-Agent-Systems in Epidemiology von [Roche u. a. \(2008\)](#), Agent-Based-Model for Spread of Dengue Fever von [Jacintho u. a. \(2010\)](#), An agent-based modeling for pandemic influenza in Egypt von [Khalil u. a. \(2010\)](#) und Vektor-born Epidemie Modell von [Erickson u. a. \(2010\)](#) dargelegt. In Kapitel 4 werden die Anforderungen für den Prototyp aufgelistet. Der Entwurf des Prototypen wird in Kapitel 5 erläutert. Das Kapitel 6 enthält die Beschreibung der Realisierung des Prototypen. In Kapitel 7 werden die Ergebnisse aus den Simulationsläufen erfasst und diskutiert. Das Kapitel 8 schließt die Arbeit mit einer Zusammenfassung und einem Ausblick auf mögliche weiterführende Arbeiten und offene Fragestellungen ab.

---

<sup>1</sup>Die Reproduktionsnummer besagt, wie viele bisher gesunde Menschen eine mit dem Erreger infizierte Person im Schnitt ansteckt (Vgl. [ETH Zürich \(2015\)](#))

## 2 Grundlagen

Die nächsten Abschnitte erläutern grundlegende Begriffe wie zum Beispiel Vektor-übertragende Krankheiten (u.a. Malaria, Denguefieber), das SIR-Epidemiemodell und Ausbreitungsmodelle.

### 2.1 Begriffserklärung

Die **Epidemiologie** untersucht die Verbreitung und Bekämpfung der Gesundheitsprobleme. Der Begriff hat folgende Bedeutung: „Die Lehre von dem, was dem Volk geschieht“.

Folgende Ziele hat die Epidemiologie:

- Erkennen der Ursachen einer Erkrankung und Ermittlung von Risikofaktoren bzw. Einflussfaktoren
- Bestimmung der Ausmaße der Erkrankung im Bezug auf die Bevölkerung
- Untersuchung des Verlaufs und der Prognose von Krankheiten
- Bewertung der Maßnahmen und Änderung des medizinischen Vorgehens
- Schaffung von Entscheidungsgrundlagen für die Gesundheitspolitik

Weitere Begriffe sind die **Endemie**, **Epidemie** und **Pandemie**, die im folgenden erläutert werden. Die **Endemie** ist ein konstantes Auftreten einer Krankheit innerhalb eines geografischen Gebiets oder einer Bevölkerungsgruppe. Die **Epidemie** beschreibt das Auftreten von mehr Krankheitsfällen in der Bevölkerung, als üblicherweise zu erwarten wären. Meist ist die Epidemie örtlich und zeitlich begrenzt. Das Infektionsschutzgesetz definiert die Epidemie wie folgt:

„Auftreten von zwei oder mehreren gleichartigen Erkrankungen, bei denen ein epidemischer Zusammenhang wahrscheinlich ist oder vermutet wird.“ (IfSG § 6, 2b und 5)

Und die **Pandemie** ist mit der Epidemie gleichzusetzen, die sich zusätzlich weltweit oder über ein weites Gebiet mit Überschreitung von internationalen Grenzen ausbreitet. Es ist eine große Anzahl von Menschen betroffen ([Ammon, A \(2012\)](#)).

## 2.2 Vektor-übertragende Infektionskrankheiten

Die nächsten Abschnitte erläutern die Vektor-übertragenen Infektionskrankheiten Dengue-Fieber, Malaria, Chikungunya-Fieber und West-Nile-Fieber.

### 2.2.1 Allgemein

Ein Moskito<sup>1</sup> kann, nach einer Infizierung, den Rest seines Lebens die Krankheit übertragen. Eine direkte Ansteckung von Mensch zu Mensch ist nicht möglich. Es gibt keine Heilung oder kommerzielle Impfstoffe für die Infektionskrankheiten Dengue-Fieber, Malaria, Chikungunya-Fieber und West-Nile-Fieber (Vgl. [World Health Organization \(2014d\)](#), [World Health Organization \(2009\)](#), [World Health Organization \(2014a\)](#), [World Health Organization \(2011\)](#)).

### 2.2.2 Dengue-Fieber

Das Dengue-Fieber ist eine Krankheit, die sich in vielen Teilen der Welt epidemisch ausbreitet. Die Ausbreitung ist bevorzugt in armen städtischen Gebieten, Vororten und auf dem Land zu finden, wirkt sich aber auch auf Nachbarschaften in tropischen und subtropischen Ländern aus. Dengue-Fieber ist eine durch Moskitos übertragene Infektionskrankheit und verursacht eine, der schweren Grippe ähnliche Erkrankung. In den letzten Jahren hat die Häufigkeit der Dengue-Fieber Erkrankungen um das 30-fache zugenommen (Vgl. [World Health Organization \(2014c\)](#)). Jährlich infizieren sich 50-100 Millionen Menschen in über 100 Ländern (u.a. Afrika, Südamerika, dem östlichen Mittelmeerraum, Südostasien und dem westlichen Pazifik), größtenteils sind Kinder betroffen. Amerika, Süd-Ost-Asien und der westliche Pazifik sind die am stärksten betroffenen Gebiete (Vgl. [World Health Organization \(2014d\)](#)). Das Denguefiebertvirus wird durch einen infizierten weiblichen Moskito (Überträger oder Vektor) auf den Menschen (Hauptopfer oder Quelle der Infektion) (durch einen Biss) übertragen (Vgl. [World Health Organization \(2014c\)](#)). Etwa 2,5% der Betroffenen sterben. Am häufigsten beißen die Mücken in den frühen Morgenstunden und am Abend vor der Dämmerung. Die Inkubationszeit beträgt etwa 4-10 Tage nach dem Biss eines infizierten Moskitos (Vgl. [World Health Organization \(2014d\)](#)). Symptome des Dengue-Fiebers sind starke Kopfschmerzen, Schmerzen hinter den Augen, Muskel- und Gelenkschmerzen, geschwollene Drüsen und Hautausschlag (Vgl. [World Health Organization \(2014g\)](#)).

---

<sup>1</sup>Der Moskito oder auch Vektor ist der Überträger einer Infektionskrankheit.

### 2.2.3 Malaria

Die Malaria ist eine tropentypische Krankheit und weltweit eine der bedeutendsten Infektionskrankheiten. Sie tritt in tropischen und subtropischen Regionen aller Kontinente - außer Australien - in etwa 100 Ländern endemisch auf. Etwa 40% der Weltbevölkerung lebt in Malaria-Endemiegebieten. Dort erkranken schätzungsweise 250 Millionen Menschen pro Jahr. Weltweit sterben jährlich 800.000 bis 1,2 Millionen Menschen an Malaria, etwa die Hälfte von ihnen sind Kinder unter fünf Jahren. Malaria wird überwiegend in Ländern Afrikas, Asiens und Südamerikas übertragen, wobei Afrika mit etwa 90% der Fälle am meisten betroffen ist (Vgl. [Robert Koch-Institut \(2012\)](#)). In der Regel erfolgt die Übertragung durch den Biss eines weiblichen Moskitos zwischen Sonnenuntergang und Sonnenaufgang (Vgl. [Robert Koch-Institut \(2012\)](#), [World Health Organization \(2009\)](#)). Die Inkubationszeit beträgt 7-15 Tage. Die Symptome der Malaria sind Fieber, Kopfschmerzen, Schüttelfrost und Erbrechen (Vgl. [World Health Organization \(2014e\)](#)). Die Krankheit kann zu schweren Komplikationen und zum Tode führen (Vgl. [World Health Organization \(2014h\)](#)).

### 2.2.4 Chikungunya-Fieber

Das Chikungunya-Fieber ist in fast 40 Ländern, auf den karibischen Inseln, Indischen- und Pazifischen Ozean, in Asien, Afrika, Europa und Amerika vertreten (Vgl. [World Health Organization \(2014a\)](#), [World Health Organization \(2014f\)](#)). Das Chikungunya-Fieber ist eine Viruserkrankung (Vgl. [World Health Organization \(2014b\)](#)) und wird durch die Bisse von infizierten weiblichen Moskitos übertragen. Die Moskitos beißen über den gesamten Tag, die Spitzen liegen in den frühen Morgenstunden und am späten Nachmittag. Nach dem Biss eines infizierten Moskitos erfolgt der Ausbruch der Krankheit in der Regel nach vier bis acht Tagen, es kann aber auch bis zu 12 Tage dauern (Vgl. [World Health Organization \(2014a\)](#)). Die Symptome sind Muskelschmerzen, Kopfschmerzen, Übelkeit, Müdigkeit und Hautausschlag. Chikungunya ist selten tödlich. Jährlich erkranken weltweit schätzungsweise 1.5 Mio. Menschen und 113 Menschen sterben am Chikungunya-Virus (Vgl. [World Health Organization \(2014b\)](#), [Guardian \(2014\)](#)).

### 2.2.5 West-Nile-Fieber

Das West-Nile-Fieber wird häufig in Afrika, Europa, dem Nahen Osten, Nordamerika und Westasien gefunden. Die Inkubationszeit beträgt gewöhnlich 3 bis 14 Tage. Der West-Nile-Fieber kann den Tod beim Menschen verursachen. Symptome sind Fieber, Kopfschmerzen, Müdigkeit und Gliederschmerzen, Übelkeit, Erbrechen, gelegentlich mit einem Hautausschlag und geschwollene Lymphdrüsen. (Vgl. [World Health Organization \(2011\)](#)). Pro Jahr erkranken

schätzungsweise 1000 Menschen und weltweit starben, seit 1999, 1200 Menschen am West-Nile-Fieber (Vgl. [MedicineNet \(2014\)](#)).

## 2.3 SIR-Modell

Das SIR-Modell ist das einfachste und bekannteste Differentialgleichungsmodell für die Beschreibung der Ausbreitung von Infektionskrankheiten (Vgl. [Khalil u. a. \(2010\)](#)). Das Modell wurde zu Beginn des 20. Jahrhunderts von Hamer, Ross und Kermack/ McKendrick entwickelt. Das SIR-Modell besteht aus den folgenden Zuständen: S, I und R, die in [Abbildung 2.1](#) zu sehen sind.



Abbildung 2.1: SIR-Modell (Entnommen aus: [Khalil u. a. \(2010\)](#))

Der Zustand S beschreibt die empfänglichen, I die infizierten und R die genesenen bzw. immunen Individuen. Die Übergänge werden durch die Pfeile zwischen den Zuständen repräsentiert. Der Übergang erfolgt nach einem Biss von einem Mosquito (Ansteckungsrate) bzw. nach der Infektionszeit (Erholungsrate). Die Gleichungen ([2.1](#), [2.2](#) und [2.3](#)), die nachfolgend zu sehen sind, entsprechen der Anzahl von Individuen in den Zuständen S, I und R in Abhängigkeit von der Zeit  $t$  (Vgl. [Khalil u. a. \(2010\)](#)).

$$\frac{dS}{dt} = -\beta SI \quad (2.1)$$

$$\frac{dI}{dt} = \beta SI - \gamma I \quad (2.2)$$

$$\frac{dR}{dt} = \gamma I \quad (2.3)$$

Die Gesamtpopulation setzt sich aus den Populationen der einzelnen Zustände zusammen.  $\beta$  beschreibt die Ansteckungsrate und  $\gamma$  die Erholungsrate. Die erste Gleichung [2.1](#) zeigt die regelmäßige Abnahme der empfänglichen Individuen. Die nächste Gleichung [2.2](#) zeigt den Zuwachs an Neuinfizierten, die den Zustand Empfänglich verlassen und dann in den Zustand Infiziert übergehen. Außerdem nimmt der Anteil der Infizierten ab, da die Infektion überwunden ist und sie gehen in den Zustand genesen bzw. immun über. Die letzte Gleichung [2.3](#) beschreibt die Zunahme der genesenen bzw. immunen Individuen (Vgl. [Khalil u. a. \(2010\)](#)).

## 2.4 Ausbreitungsmodelle

Der nächste Abschnitt beschreibt die Ausbreitungsmodelle in Form eines agentenbasierten Modells und zellulären Automaten.

### 2.4.1 Agentenbasiertes Modell

Das agentenbasierte Modell besteht aus den folgenden drei Elementen: Agenten, Agentenbeziehungen und einer Umwelt, die in der Abbildung 2.2 zu sehen sind.

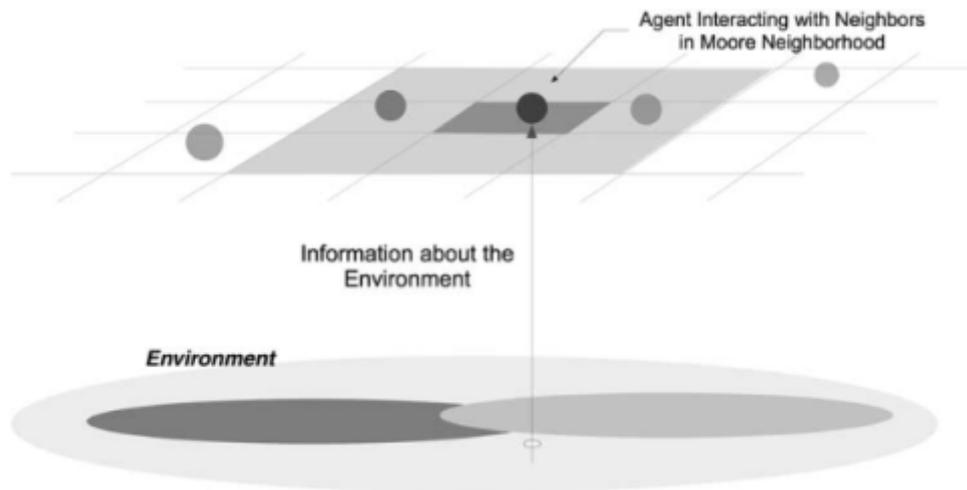


Abbildung 2.2: Agentenbasiertes Modell

(Entnommen aus: [Macal und North \(2010\)](#))

Jeder **Agent** hat Attribute (Autonomie<sup>2</sup>, Modularität<sup>3</sup>, Sozialität<sup>4</sup> und Konditionalität<sup>5</sup>) und Verhaltensweisen. Agenten leben in und mit ihrer **Umwelt** und interagieren mit anderen Agenten. Agenten haben **Agentenbeziehungen**, die durch die Methode der Interaktion und der Topologie festgelegt wird.

Die Agentenbeziehungen beschreiben die Interaktionen eines Agenten, wer er ist oder wer er sein könnte und mit wem er interagiert. Die Abbildungen 2.3 (a-e) zeigen die verschiedenen Topologien, nach denen ein Agent interagieren kann.

---

<sup>2</sup>Ein Agent ist autonom und steuert sich selbst

<sup>3</sup>Ein Agent ist eigenständig

<sup>4</sup>Ein Agent ist sozial und interagiert mit anderen Agenten

<sup>5</sup>Ein Agent hat einen Zustand

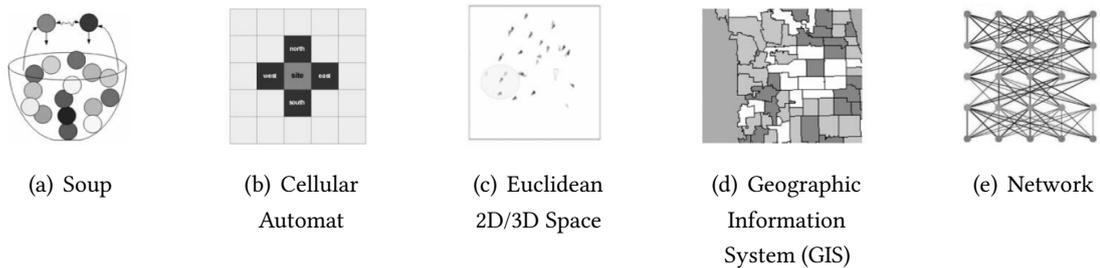


Abbildung 2.3: Agentenbasiertes Modell - Topologien  
(Entnommen aus: [Macal und North \(2010\)](#))

Die Abbildung [2.3\(a\)](#) zeigt die Topologie Suppe, die ein nicht räumliches Modell ist, in dem Agenten keine Attribute haben und dadurch keinen Standortvorteil haben. Die nächste Abbildung [2.3\(b\)](#) zeigt einen zellulären Automaten, nach dem Agenten interagieren können. Die Topologie stellt den Agenten ein Interaktionsmuster und verfügbare Informationen im Gitter bereit. Die Zellen in unmittelbarer Umgebung sind die Nachbarn des Agenten. Die mittlere Abbildung [2.3\(c\)](#) zeigt, dass die Agenten sich in 2D oder 3D Räumen bewegen. Die Abbildung [2.3\(d\)](#) zeigt ein Geografisches Informationssystem in dem sich die Agenten über realistische Geodaten-Landschaften bewegen. Die letzte Abbildung [2.3\(e\)](#) zeigt ein Netzwerk, in dem die Verbindungen untereinander statisch, Links werden vorgegeben, oder dynamisch, Links werden vom Netzwerk bestimmt, bestehen.

Die Topologien werden verwendet, um Agenten zu verbinden und sie schafft die Grundlage für Agentenbeziehungen. Die Agenten interagieren zu bestimmten Zeitpunkten mit einer Menge von Agenten in der Umwelt (Vgl. [Macal und North \(2010\)](#), [Macal und North \(2011\)](#)).

### 2.4.2 Zellulärer Automat

Die Abbildung [2.4](#) zeigt einen zellulären Automaten. Ein zellulärer Automat besteht aus Zellen, einem Zellraum, einem Zellstatus, einer Liste der angrenzenden Zellen (Nachbarn) und den Übergangsregeln. Die Elemente werden im Folgenden detaillierter erläutert.

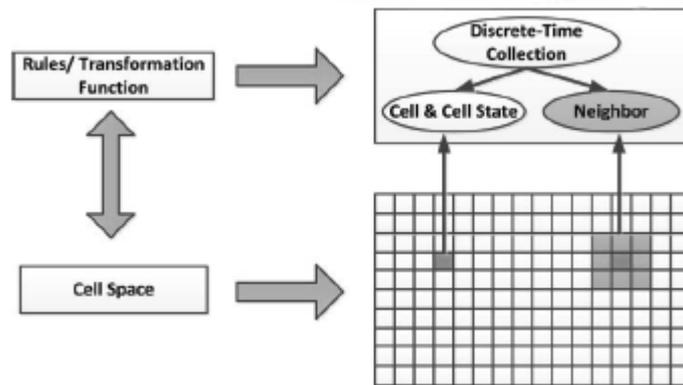


Abbildung 2.4: Zellulärer Automat

(Entnommen aus: Sun u. a. (2012))

Die **Zelle** ist das grundlegende Element im Modell. Der Einfluss einer Zelle wird durch Parameter festgelegt, außerdem kann die Anzahl der Zellmitglieder, die eine Zelle halten kann, bestimmt werden. Ein zweidimensionales Array aus Zellen beschreibt den **Zellraum**. Die Form des Zellraums kann unterschiedlich definiert werden. Die Position einer Zelle liegt im zweidimensionalen Array und wird durch die  $X$ - und  $Y$ -Koordinate beschrieben. Der **Zellstatus** beschreibt den Zustand einer Zelle zum Zeitpunkt  $t$  im zellulären Automaten. Die direkten Nachbarn, die an die Zelle angrenzen, werden in der **Liste der angrenzenden Zellen** gepflegt. Ein Zustandswechsel ist abhängig vom eigenen Zustand und dem Zustand der Nachbarzellen und wird mit den **Übergangsregeln** festgelegt (Vgl. Sun u. a. (2012)).

## 3 Verwandte Arbeiten

Das Kapitel zeigt einen Einblick in die Arbeiten Multi-Agent-Systems in Epidemiology (3.1) von Roche u. a. (2008), Agent-Based-Model for the Spread of the Dengue Fever (3.2) von Jacintho u. a. (2010), An agent-based modeling for pandemic influenza in Egypt (3.3) von Khalil u. a. (2010) und Vector-born Epidemie Modell (3.4) von Erickson u. a. (2010).

### 3.1 Multi-Agent-Systems in Epidemiology

Die Arbeit von Roche u. a. (2008) zeigt eine Klassifizierung in Vektor (Moskito) und Reservoir<sup>1</sup> (Mensch). Der Vektor ist der Überträger einer Infektionskrankheit. Der Erreger überlebt im Körper des Vektors ohne diesen Körper zu befallen, z.B. bei blutsaugende Insekten.

#### 3.1.1 SEIR-Epidemiemodell

In Abbildung 3.1 ist das SEIR-Epidemiemodell zu sehen.



Abbildung 3.1: SEIR-Epidemiemodell

(In Anlehnung an: Roche u. a. (2008))

Das Modell zeigt folgende vier Zustände: **S**usceptible (Anfällig), **E**xposed (Infiziert), **I**nfectious (Infektiös) und **R**ecovered (Geheilt). Das Individuum wird im Zustand *Anfällig* geboren. Nach einer Infektion wechselt das Individuum in den Zustand *Infiziert*, in diesem Zustand ist das Individuum unfähig andere Individuen anzustecken. Nach einer Latenzzeit geht das Individuum in den Zustand *Infektiös* über und ist nun in der Lage andere Individuen anzustecken. Nach der

<sup>1</sup>Erregerträger oder Wirt, der einen Erreger in sich trägt

Infektionsperiode geht der Individuum in den Zustand *Geheilt* über und ist nun immun gegen die Krankheit. Eine Ausnahme besteht bei Vektoren, diese bleiben bis zu ihrem Tod infektiös.

### 3.1.2 Modell

Die Abbildung 3.2 zeigt die Komponenten des Multi-Agent-Systems.

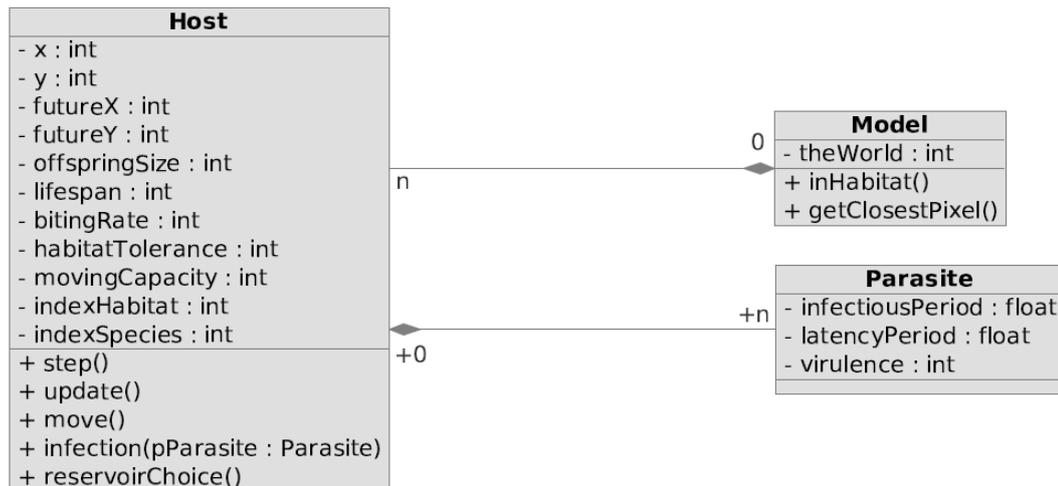


Abbildung 3.2: Komponenten des Multi-Agenten-Systems  
(Entnommen aus: Roche u. a. (2008))

Die Parasit-Komponente hat die Eigenschaften *Infektiöse Periode* (*infectiousPeriod*), *Latenzperiode* (*latencyPeriod*) (Inkubationszeit) und *Virulenz* (*virulence*). Virulenz ist die Aggressivität eines Krankheitserregers eine Krankheit auszulösen. Die Eigenschaften können sowohl vom Vektor, wie auch vom Reservoir angewendet werden.

Das **Modell** (theWorld) wird durch ein Array abgebildet, das die Größe der virtuellen Welt darstellt. Das Array enthält Integer-Werte und ist verknüpft mit dem Index Lebensraum in der Host Komponente. Das Array charakterisiert die räumliche Verteilung der Host-Arten und die Adressierung jeder Landschaftskonfigurationen.

Der **Host** stellt die wichtigste Komponente im Modell dar. Die Komponente hat verschiedene Gruppen von Eigenschaften. Die erste Gruppe betrifft die geografische Lokalisierung von Personen und Moskitos (Parasite). Der Host enthält *x*- und *y*-Attribute, die seine geografische Position darstellen. Für Updatezwecke enthält der Host auch noch die *zukünftigen x*- und

y-Attribute, die die zukünftige Position darstellen. Diese Eigenschaften werden im Laufe der Zeit dynamisch aktualisiert. Die zweite Gruppe enthält Eigenschaften, die nicht im Laufe der Zeit aktualisiert werden. Folgende Eigenschaften enthält die Host Komponente: *Nachkommengröße*, *Lebensdauer*, *Beißrate*, Räumliche Merkmale, *Index Lebensraum* und *Index Art*. Die *Beißrate* unterscheidet sich bei Vektor und Reservoir. Ein Reservoir hat eine Beißrate von -1 und der Vektor eine entsprechend definierte Beißrate. Zu den räumlichen Merkmalen zählen *Bewegungskapazität* und *Lebensraumtoleranz*. Die Bewegungskapazität stellt einen Bereich dar, indem sich ein Host bewegen darf. Die Lebensraumtoleranz gibt an, wie weit sich der Host außerhalb seines ursprünglichen Lebensraum bewegen darf. *Index Lebensraum* und *Index Art* werden für rechnerische Zwecke verwendet. Alle Individuen einer bestimmten Host-Art haben die gleiche Index Art und sie können in verschiedenen Index Lebensräumen vorkommen. Die Host-Reaktionen gegen den Parasiten wurden nicht berücksichtigt. Die Interaktion zwischen Host und Parasit sind statisch in der Parasit-Komponente festgelegt.

#### 3.1.3 Komponentenzusammenspiel

##### Host mit Model

Die Funktion **move** ist die grundlegendste Bewegung, jeder einzelne Host hat eine Bewegungskapazität, sie erlaubt dem Host sich von einer Ausgangsposition [(Bewegungskapazität +1) \*2] zu einer neuen Position zu bewegen. Dies gilt sowohl für den Vektor als auch für den Reservoir. Die nächste Bewegung erfolgt zufällig im Quadrat, ist diese Bewegung nicht im Lebensraum inklusive der Lebensraumtoleranz wird die nächste gültige Position im Lebensraum ausgewählt. Der Vektor hat zusätzlich noch eine Beißrate. Die Wahrscheinlichkeit des Beißens steigt mit jedem Schritt [1/Beißrate]. Die Anzahl von verfügbaren Reservoirpositionen wird errechnet. Die Reservoir sind gleichmäßig im Modell verteilt. Ein Vektor wählt eine Position von errechneten Positionen und beißt den Reservoir. Daraus folgt eine potenzielle Übertragung der Krankheit zwischen Vektor und Reservoir.

##### Host - Geburt und Tod

Jedes Individuum hat eine Nachkommensgröße, sie repräsentiert die Anzahl von neuen Hosts, die nach der Geburt des Hosts erstellt werden. Die Nachkommen werden verteilt über seine Lebensdauer erzeugt und die Fähigkeit sinkt mit dem Alter. Um eine konstante Population zu erreichen, wurde die Nachkommensgröße auf 1 gesetzt. Der Host bekommt einen Nachkommen, bevor er stirbt, somit bleibt das Gleichgewicht bestehen.

### 3.1.4 Funktionen

Die **step**-Funktion bewegt den Vektor und den Reservoir auf die nächste Position, sie kann erst aufgerufen werden, wenn die move-Funktion (siehe Abschnitt **Host mit Model**) vom Vektor und Reservoir aufgerufen worden ist. Zusätzlich hat der Vektor noch ein Bite-Rate, d.h. der Vektor sucht nach verfügbaren Reservoiren und beißt einen von ihnen. In diesem Fall können die 3 Felder FutureX, FutureY und nextParasite modifiziert werden.

Nach der step-Funktion wird bei allen Individuen die **update**-Funktion aufgerufen. Einige Felder werden aktualisiert, wie X und Y durch FutureX und FutureY, currentParasite durch futureParasite und pastParasite durch currentParasite. Zusätzlich werden noch die birth und dead Funktionen aufgerufen (siehe **Host - Geburt und Tod**).

Außerdem hat das Modell noch 2 weitere Funktionen, **inHabitat** gibt an, ob der Host sich noch im Lebensraum befindet und **getClosestPixel** gibt die nächste gültige Position aus dem Modell zurück. Zudem hat der Host die Funktion **reservoirChoice**, die die Reservoiren zurück gibt, die sich in der näheren Umgebung befinden.

## 3.2 ABM<sup>2</sup> for the Spread of the Dengue Fever

Die Arbeit von [Jacintho u. a. \(2010\)](#) zeigt die Eindämmung einer Epidemie durch einen Exterminator (Schädlingsbekämpfer) und die Wiederinfizierung<sup>3</sup> eines Agenten durch einen Moskito.

### 3.2.1 Modell

In Abbildung 3.3 ist das Komponentenmodell zu sehen.

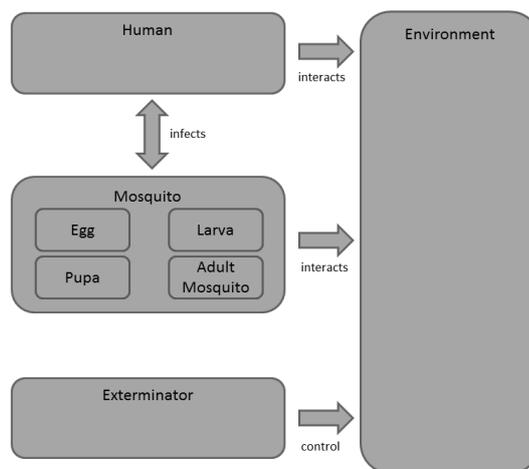


Abbildung 3.3: Komponenten des Agent-based Model  
(In Anlehnung an: [Jacintho u. a. \(2010\)](#))

Bei mehrfacher Wiederinfizierung steigt die Wahrscheinlichkeit der Infizierung und die Sterberate beim **Human**. Bei 2 Infizierungen beträgt die Sterberate 0,5%, bei drei 10%, bei vier 15% und bei mehr als vier 25%.

Der **Exterminator** (Schädlingsbekämpfer) bewegt sich in der Umwelt zu Gebieten mit einer sehr hohen Anzahl an Mücken. Die Organisation des Gesundheitswesens stellt die Karten zur Verfügung und gibt die gefährdeten Gebiete an, bei denen Kontrollmaßnahmen geplant sind. Die Aufgabe des Schädlingsbekämpfers ist es, alle erwachsenen Moskitos zu töten. Ein Schädlingsbekämpfer ortet eine Menge von erwachsenden Moskitos, dann sprüht er sein Gift über eine Fläche aus. Die Beschilderung erfolgt durch Sensoren, sie geben die Konzentration

---

<sup>2</sup>Agent-Based-Model

<sup>3</sup>Erneute Infizierung nachdem der Agent genesen ist

der Moskitos in den Gebieten an. Bei höher Konzentration verfärbt sich das Gebiet auf der Karte dunkel. Sinkt die Konzentration im Gebiet erfolgt eine Abstufung der Farbe auf der Karte.

### 3.3 An agent-based modeling for pandemic influenza in Egypt

Die Arbeit von [Khalil u. a. \(2010\)](#) beschreibt die Eindämmung einer Infektionskrankheit. Nachdem ein Individuum infiziert wurde und es sich in der Nähe eines Doktors befindet, wird das Individuum unter Quarantäne gestellt. Im besten Fall tritt die Genesung ein oder im schlimmsten Fall verstirbt das Individuum.

### 3.4 Vector-born Epidemie Modell

Die Arbeit von [Erickson u. a. \(2010\)](#) beschreibt das Host-Vektor Modell, dass eine Erweiterung des SEIR-Epidemiomodells, um den Vektor (Moskito) ist, wie es in [Beier \(2013b\)](#) beschrieben wurde.

#### 3.4.1 Modell

Die Abbildung 3.4 zeigt das Host-Vektor-Modell.

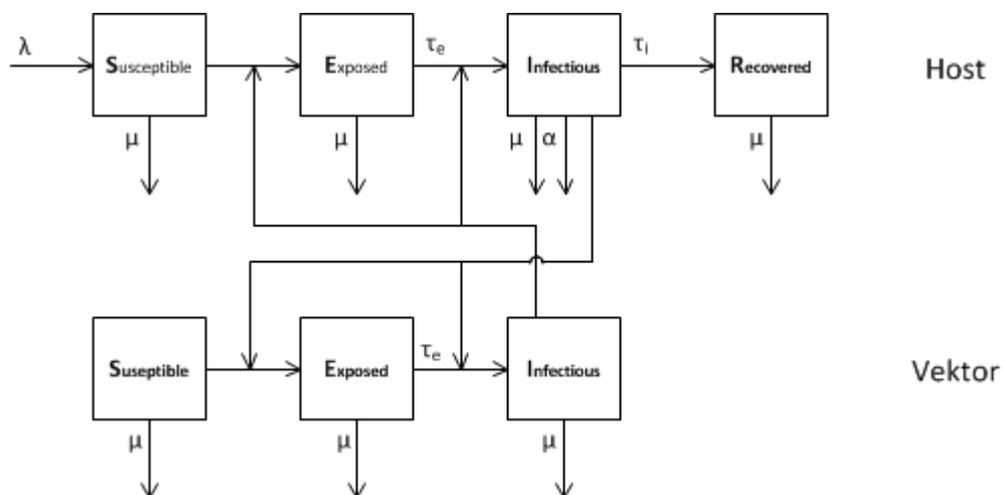


Abbildung 3.4: Host-Vektor Modell

(In Anlehnung an: [Erickson u. a. \(2010\)](#))

Das Host-Vektor-Modell besteht aus zwei Epidemiemodellen, eins für den Host und eins für den Vektor. Mit dieser Erweiterung ist es auch möglich, die Verbindung zwischen Host und Vektor zu zeigen.

Das Host-Modell zeigt folgende vier Zustände: Susceptible (Anfällig), Exposed (Infiziert), Infectious (Infektiös) und Recovered (Geheilt). Das Individuum wird im Zustand *Anfällig* geboren. Nach einer Infektion wechselt das Individuum in den Zustand *Infiziert*, in diesem Zustand ist das Individuum unfähig Vektoren anzustecken. Nach einer Latenzzeit geht das Individuum in den Zustand *Infektiös* über und ist nun in der Lage Vektoren anzustecken. Nach der Infektionsperiode geht der Individuum in den Zustand *Geheilt* über und ist nun immun gegen die Krankheit.

Im Gegensatz zu dem Host-Modell, hat das Vektor-Modell nur drei Zustände: Susceptible (Anfällig), Exposed (Infiziert) und Infectious (Infektiös). Der Vektor wird im Zustand *Anfällig* oder im Zustand *Infiziert*, bei Übertragung der Viren durch das Muttertier, geboren. Nach einer Infektion wechselt der Vektor in den Zustand *Infiziert* und der Vektor ist unfähig Individuen anzustecken. Nach einer Latenzzeit geht der Vektor in den Zustand *Infektiös* über und ist nun in der Lage Individuen anzustecken. Nach der Infektionsperiode bleibt der Vektor im Zustand *Infektiös* bis der Vektor verstirbt.

Folgende **Parameter**, mit Unterteilung in Host und Moskito, sind gegeben und werden in den nachfolgenden Formeln verwendet:

c: Kontaktrate

*Moskito Population (v):*

$S_v$ : Anzahl an anfälligen (susceptible) Moskitos

$E_v$ : Anzahl an infizierten (exposed) Moskitos

$I_v$ : Anzahl an infektiösen (infectious) Moskitos

$T_v$ : Gesamtanzahl an Moskitos

$\tau e_v$ : Übertragungsrate Susceptible (Anfällig)  $\rightarrow$  Exposed (Infiziert)

$\mu_v$ : Natürliche Sterberate

*Host Population (h):*

$S_h$ : Anzahl an anfälligen (susceptible) Hosts

$E_h$ : Anzahl an infizierten (exposed) Hosts

$I_h$ : Anzahl an infektiösen (infectious) Hosts

$R_h$ : Anzahl an geheilten (recovered) Hosts

$T_h$ : Gesamtanzahl an Hosts

$\lambda_h$ : Geburtenrate (wird nicht berücksichtigt, da die Population konstant gehalten wird)

$\tau e_h$ : Übertragungsrate Susceptible (Anfällig)  $\rightarrow$  Exposed (Infiziert)

$\tau i_h$ : Infektionsrate Exposed (Infiziert)  $\rightarrow$  Infectious (Infektiös)

$\alpha_h$ : Durch Krankheit versuchte Todesrate

$\mu_h$ : Natürliche Todesrate

Die dazugehörigen **Gleichungen**, mit denen sich die aktuelle Population, je nach Zustand, berechnen lässt, werden nachfolgend gezeigt.

Für den Host das **SEIR-Hostmodell**:

$$\frac{dS_h}{dt} = \lambda_h * T_h - S_h * \left( \frac{c * I_v}{T_h} + \mu_h \right) \quad (3.1)$$

$$\frac{dE_h}{dt} = S_h * \frac{c * I_v}{T_h} - E_h * (\tau e_h + \mu_h) \quad (3.2)$$

$$\frac{dI_h}{dt} = E_h * \tau e_h - I_h * (\tau i_h + \alpha + \mu_h) \quad (3.3)$$

$$\frac{dR_h}{dt} = I_h * \tau i_h - \mu_h * R_h \quad (3.4)$$

$$T_h = S_h + E_h + I_h + R_h \quad (3.5)$$

Für den Vektor das **SEI-Vektormodell**:

$$\frac{dS_v}{dt} = \mu_v * T_v - S_v * \left( \frac{c * I_h}{T_h} + \mu_v \right) \quad (3.6)$$

$$\frac{dE_v}{dt} = S_v * \frac{c * I_h}{T_h} - E_v * (\tau e_v + \mu_v) \quad (3.7)$$

$$\frac{dI_v}{dt} = E_v * \tau e_v - \mu_v * I_v \quad (3.8)$$

$$T_v = S_v + E_v + I_v \quad (3.9)$$

Ob es sich um eine Epidemie handelt, lässt sich über die Reproduktionsnummer bestimmen. Ist der berechnete Wert größer als 1, handelt es sich um eine Epidemie.

Für den Host errechnet sich die **Reproduktionsnummer** mit

$$R_0^{HV} = \frac{c * \tau e_h}{(\tau e_h + \mu_h) * (\tau i_h + \alpha + \mu_h)} \quad (3.10)$$

Für den Vektor errechnet sich die **Reproduktionsnummer** mit

$$R_0^{VH} = \frac{c * T_v * \tau e_v}{T_h * \mu_v * (\tau e_v + \mu_v)} \quad (3.11)$$

Die zusammengefasste **Reproduktionsnummer** setzt sich aus den Host und Moskito Reproduktionsnummern zusammen.

$$R_0 = \sqrt{R_0^{HV} * R_0^{VH}} \quad (3.12)$$

## 4 Analyse

Die nächsten Abschnitte zeigen die Anforderungen des Prototyps, die zum Teil aus [Beier \(2013b\)](#) stammen.

### 4.1 Anforderungen

Im folgenden werden die Anforderungen vom Mensch (Reservoir), Moskito (Vektor), Umwelt, Konfiguration und Darstellung, sowie die technischen Anforderungen erläutert.

#### 4.1.1 Agent (Reservoir)

- A1 Der Agent hat eine Identifizierungsnummer, einen Zustand, eine aktuelle Lebenszeit, einen aktuellen Standort\*, einen zukünftigen Standort\*, eine Lebensdauer\*, eine Nachkommensgröße\*, eine Kategorie (Agent oder Schädlingsbekämpfer) und eine Eigenschaft\* die anzeigt, ob der Agent von einem Moskito gebissen worden ist (\*nach [Roche u. a. \(2008\)](#)).
- A2 Der Agent ändert seinen Zustand in *Infiziert*, nachdem er von einem infiziertem Moskito gebissen wurde (nach [Roche u. a. \(2008\)](#)).
- A3 Nach einer Inkubationszeit ändert sich der Zustand von *Infiziert* in *Infektiös* und der Agent befindet sich in der Infektionsperiode. In dieser Zeit ist der Agent in der Lage gesunde Moskitos zu infizieren (nach [Roche u. a. \(2008\)](#)).
- A4 Nach der Infektionsperiode ändert sich der Zustand des Agenten in *Geheilt*. Von nun an ist der Agent geheilt und immun gegen die Krankheit (nach [Roche u. a. \(2008\)](#)).
- A5 Wiederinfizierungen sind möglich. In diesem Fall entfällt der Zustand *Geheilt*.
- A6 Der Agent bewegt sich frei durch die Umwelt (nach [Roche u. a. \(2008\)](#)).
- A7 Der Agent kann Nachkommen bekommen. Nachdem der Agent gestorben ist, wird ein neuer Nachkommen, im Zustand (Anfällig<sup>1</sup> oder Infiziert<sup>2</sup>) des verstorbenen Agenten, erstellt. Hierdurch bleibt die Population konstant (nach [Roche u. a. \(2008\)](#)).

---

<sup>1</sup>Der verstorbene Agent befand sich vor seinem Tod im Zustand Anfällig oder Geheilt

<sup>2</sup>Der verstorbene Agent befand sich vor seinem Tod im Zustand Infiziert oder Infektiös

- A8 Der Agent kann in Quarantänestationen aufgenommen werden, wenn er sich in der Reichweite einer Quarantänestation befindet. In der Quarantänestation werden äußere Einflüsse (Moskitos) ferngehalten, so dass der Agent die Chance hat zu überleben.
- A9 Ist der Agent der Kategorie Schädlingsbekämpfer zugeordnet, wird der Agent wöchentlich aktiv und tötet im virtuellen Feld befindliche Moskitos.
- A10 Zum Simulationsstart wird eine Reproduktionsnummer ( $R$ )<sup>3</sup> errechnet.

### 4.1.2 Moskito (Vektor<sup>4</sup>)

- M1 Der Moskito hat eine Identifizierungsnummer, einen Zustand, eine aktuelle Lebenszeit, einen aktuellen Standort\*, einen zukünftigen Standort\*, eine Nachkommensgröße\* und eine Lebensdauer\* (\*nach Roche u. a. (2008)).
- M2 Der Moskito wird im Zustand Anfällig oder Infiziert geboren. Nach einer Infektion mit dem Virus bleibt der Moskito sein Leben lang im Zustand Infektiös (nach Roche u. a. (2008)).
- M3 Der infizierte Moskito ändert seinen Zustand in Infiziert, wenn er einen Agenten gebissen hat, der sich in dem Zustand Infektiös befand (nach Roche u. a. (2008)).
- M4 Nach einer Inkubationszeit ändert sich der Zustand von Infiziert in Infektiös und der Moskito befindet sich in der Infektionsperiode. In dieser Zeit ist der Moskito in der Lage, durch einen Biss, einen gesunden Agenten zu infizieren. (nach Roche u. a. (2008)).
- M5 Der Moskito bewegt sich frei durch die Umwelt (nach Roche u. a. (2008)).
- M6 Der Moskito kann Nachkommen bekommen. Nachdem der Moskito gestorben ist, wird ein neuer Nachkomme, im Zustand (Anfällig<sup>5</sup> oder Infiziert<sup>6</sup>) des verstorbenen Moskitos, erstellt. Hierdurch bleibt die Population konstant.
- M7 Der Moskito hat zusätzlich noch die Fähigkeit ein virtuelles Feld abzufragen und dort einen Agenten auszuwählen, um diesen zu beißen (nach Roche u. a. (2008)).

### 4.1.3 Umwelt

- U1 In der Umwelt bewegen sich alle Agenten und Moskitos (nach Roche u. a. (2008)).

---

<sup>3</sup>Gibt an, ob es sich um eine Epidemie handelt ( $R < 1 < R$ , ist der Wert größer gleich 1, dann handelt sich um eine Epidemie, ansonsten nicht)

<sup>4</sup>Überträger einer Infektionskrankheit, die Erreger überleben im Körper im aktiven Zustand, befallen den Körper aber nicht

<sup>5</sup>Der verstorbene Agent befand sich vor seinem Tod im Zustand Anfällig

<sup>6</sup>Der verstorbene Agent befand sich vor seinem Tod im Zustand Infiziert oder Infektiös

- U2 Der Umwelt können Koordinaten übergeben werden. Durch die Liste von Koordinaten kann z.B. die Umwelt auf ein Land oder Quarantänestationen abgebildet werden.
- U3 Die Agenten und Moskitos können sich nur innerhalb dieser Umwelt bewegen (nach Roche u. a. (2008)).
- U4 Die Umwelt gibt nach Anfrage vom Agent bzw. Moskito ein virtuelles Feld zurück, das Agenten bzw. Moskitos enthält.
- U5 Die Agenten und Moskitos haben eine Position, die durch eine x- und y-Koordinate angegeben wird (nach Roche u. a. (2008)).
- U6 Jeder Durchlauf wird aufgezeichnet und nach dem SEIR-Epidemiemodell dargestellt.

### 4.1.4 Konfiguration

Über eine Konfigurationsdatei sind folgende Eigenschaften zu beeinflussen:

- K11 Populationsgröße von Agenten im Zustand Anfällig
- K12 Populationsgröße von Moskitos im Zustand Anfällig und Infiziert
- K13 Lebensdauer von Agenten und Moskitos
- K14 Anzahl von Nachkommen von Agenten und Moskitos
- K15 Sterberate von Agenten und Moskitos
- K16 Übertragungsrate von Agenten und Moskitos
- K17 Sterberate von Agenten, beeinflusst durch die Krankheit
- K18 Geburtenrate von Agenten
- K19 Infektionsrate von Agenten
- K20 Kontaktrate von Agenten und Moskitos
- K21 Koordinaten der Umwelt und Quarantänestationen
- K22 Wiederinfizierung von Agenten durch einen Moskito

### 4.1.5 Darstellung

- D1 Die Benutzeroberfläche zeigt einen geografischen Bereich, der durch Koordinatengrenzen bestimmt wird, an.

D2 Moskitos werden durch schwarze (Anfällig), blaue (Infiziert) und violette (Infektiös) Punkte und Agenten durch grüne (Anfällig), gelbe (Infiziert), rote (Infektiös) und graue (Geheilt) Punkte dargestellt. Die Farben der Agenten spiegeln die Zustände des SEIR-Epidemiomodells wieder und die Farben der Moskitos stellen die Zustände des SEI-Epidemiomodells dar.

### 4.1.6 Technische Anforderungen

- T1 Die Anwendung wird auf dem Betriebssystem Windows 7 Pro entwickelt.
- T2 Das System wird mit der Entwicklungsumgebung Visual Studio 2012, mit dem Framework .NET 4.5 und der Programmiersprache C# entwickelt.
- T3 Die Oberfläche der Anwendung wird mit dem Framework Windows Presentation Foundation (WPF) entwickelt.

# 5 Entwurf

Die nächsten Abschnitte erläutern die Architektur und den Ablauf des Prototyps, die zum Teil aus [Beier \(2013b\)](#) entnommen wurde.

## 5.1 Architektur

Die folgende Abbildung 5.1 zeigt die Quasar-Architektur<sup>1</sup> des Prototypen, der aus dem Anwendungskern, der Simulationsoberfläche, einer Fassade, einem Adapter und dem Dateisystem besteht.

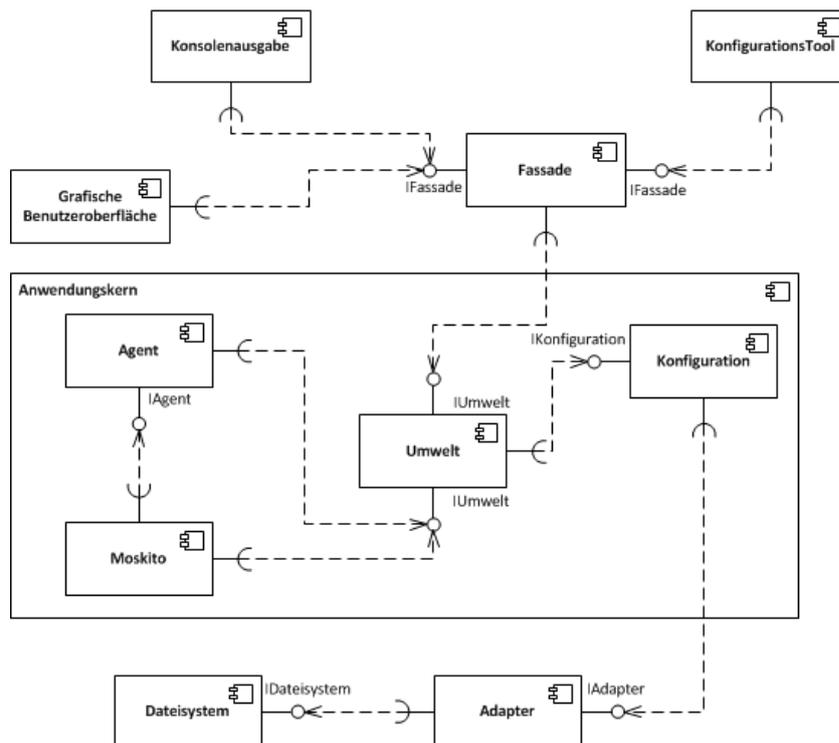


Abbildung 5.1: Architektur des Prototypen

<sup>1</sup>Die Quasar-Architektur besteht aus A-Komponenten (Anwendungskern), T-Komponenten (GUI) und TI-Komponenten (technische Infrastruktur) ([Kim Beier, 2012](#)).

Die Klassen Umwelt, Zustände, Kategorien, EpidemieModell und Lokation wurden in der Komponente Umwelt zusammengefasst. Agent und Moskito werden jeweils als eigene Komponente betrachtet.

### 5.1.1 Anwendungskern

Im Weiterem werden die Komponenten Umwelt, Agent, Moskito und Konfiguration des Anwendungskerns erläutert.

#### Umwelt

Die Komponente Umwelt ist der begrenzende Lebensraum, indem sich Moskitos und Agenten frei bewegen können. Die Umwelt verwaltet die Klasse Lokation, die neue Positionen berechnet. Außerdem werden die Aufzählungen (Enumerationen) der möglichen Zustände (None, Susceptible, Exposed, Infectious, Recovered und Dead) und Kategorien (Agent und Exterminator (Schädlingsbekämpfer)) zur Verfügung gestellt. Ferner enthält die Komponente Umwelt noch die Klasse EpidemieModell, die einen Graphen über die Verteilung der Anzahl und Zustände der Agenten eines Simulationsdurchlaufs erstellt und die Reproduktionsnummer berechnet. Außerdem stellt die Klasse Umwelt die Methoden *InVirtualField()* und *InEnvironment()* bereit, die von den Komponenten Agent und Moskito verwendet werden. Die Methode *InVirtualField()* gibt alle Agenten bzw. Moskitos zurück, die sich in einer definierten Nähe eines Moskitos bzw. Agents befinden. Und die Methode *InEnvironment()* gibt an, ob der Agent bzw. Moskito sich noch in der Umwelt befindet.

#### Agent

Der Agent bewegt sich zufällig und schrittweise durch die Umwelt. Nachdem der Agent durch einen infektiösen Moskito gebissen wurde, ändert sich der Zustand des Agenten. Beißt ein gesunder Moskito einen infektiösen Agenten, ändert sich der Zustand des Moskitos von Anfällig in Infiziert. In regelmäßigen Abständen ändert sich der Zustand des Agenten nach dem SEIR-Epidemiemodell und zusätzlich wird die Lebensdauer abgefragt, wenn sie abgelaufen ist, wird der Agent entfernt und ein neuer erstellt.

#### Moskito

Der Moskito bewegt sich zufällig schrittweise in der Umwelt. Nach der Abfrage seines virtuellen Feldes wählt der Moskito zufällig einen Agenten im gleichen virtuellen Feld und beißt diesen. Beißt der Moskito einen infektiösen Agenten, wird der Zustand eines Moskitos im

Zustand Anfällig zum Zustand Infiziert gewechselt. Außerdem wird regelmäßig der aktuelle Zustand abgefragt und entsprechend dem SEI-Epidemiemodells Änderungen durchgeführt. In regelmäßigen Abständen wird das Alter des Moskito abgefragt, der Moskito wird entfernt, wenn die Lebensdauer abgelaufen ist und ein neuer Moskito wird im Zustand des verstorbenen Moskito erstellt.

### **Konfiguration**

In der Konfiguration können die Werte einzelner Parameter (Moskito Death Rate, Moskito Lifespan, Moskito Children Size, Moskito Transmission Rate, Moskito Population Susceptible, Moskito Population Infectious, Human Death Rate, Human Death Rate Disease, Human Growth Rate, Human Lifespan, Human Children Size, Human Infection Rate, Human Transmission Rate, Human Population Susceptible, Contact Rate, Koordinaten List Hospital, Coordination List Environment) eingestellt werden. Zusätzlich bietet die Klasse die Möglichkeit, die Parameterwerte in eine XML-Datei zu schreiben und wieder auszulesen.

#### **5.1.2 Fassade**

Die Fassade stellt Methoden bereit, die die Benutzeroberfläche, Consolenausgabe und KonfigurationsTool benötigen. Die folgenden Methoden stellt die Fassade bereit:

- Factorymethoden zum Erzeugen von Agenten und Moskito
- Zugriffe auf die Methode *Run* der Komponente Agent und Moskito
- Zugriffe auf die Methode *Write* und *Read* der Konfiguration Komponente
- Zugriffe auf die Methode *Get* der Umwelt Komponente

#### **5.1.3 Benutzeroberflächen**

Die nächsten Abschnitte erläutern die Benutzeroberflächen Grafische Benutzeroberfläche, Consolenausgabe und KonfigurationsTool.

##### **Grafische Benutzeroberfläche**

Auf der grafischen Benutzeroberfläche werden die Agenten und Moskito mit der aktuellen Position und dem aktuellen Zustand dargestellt. Zusätzlich gibt es die Möglichkeit, die Simulation über die Schaltflächen Start, Stop und Pause zu steuern. Außerdem ist es möglich, das KonfigurationsTool zur Laufzeit direkt aufzurufen, um Parametereinstellungen zu ändern.

Nach der Änderung werden alle Werte angezeigt. Die grafische Benutzeroberfläche bietet nicht die Möglichkeit, wie die Konsolenausgabe (siehe Abschnitt 5.1.3), die Simulation mit einer vordefinierter Anzahl von Simulationsschritten durchzuführen, sowie die Ausgabe der Ergebnissen des Simulationsdurchlaufs in eine Datei. Bei einer hohen Anzahl von Agenten wird die grafische Benutzeroberfläche unübersichtlich und nicht konsistent dargestellt, deswegen wurde die Benutzeroberfläche Konsolenausgabe entwickelt.

### **Konsolenausgabe**

Die Konsolenausgabe bietet die Möglichkeit einen Simulationsdurchlauf, ohne grafische Benutzeroberfläche, mit ein definierten Anzahl von Simulationsschritten durchzuführen. Am Ende der Simulation wird eine Datei geschrieben, die die Ergebnisse des Simulationsdurchlaufs enthält.

### **KonfigurationsTool**

Mit dem Konfigurationstool können die Werte der Parameter einstellt bzw. geändert werden. Zusätzlich bietet die Oberfläche des Konfigurationstools die Möglichkeit die Grenzen der Umwelt und Quarantänestationen visuell festzulegen. Zum Abschluss werden die Werte gespeichert, d.h. in eine XML Datei geschrieben.

#### **5.1.4 Adapter**

Der Adapter ist ein weiterer Bestandteil der Architektur, der als Bindeglied zwischen dem Anwendungskern und dem Nachbarsystem Dateisystem dient.

#### **5.1.5 Dateisystem**

Das Dateisystem bietet eine Persistenzschicht, in der auf Dateien auf einer Festplatte zugegriffen wird.

## 5.2 Laufzeitsicht

In Abbildung 5.2 (siehe nächste Seite) ist die Laufzeitsicht der Anwendung zu sehen. Von der Benutzeroberfläche aus werden die Werte der Eigenschaften in eine XML-Datei geschrieben (*Write(Dictionary<Location> CoordinationListHospital, Dictionary<Location> CoordinationListEnvironment, List<double> MoskitoValues, List<double> humanValues, double contactRate)*) und wieder gelesen (*Read()*). Durch diesen Vorgehen kann die Benutzeroberfläche oder das Konfigurationstool die Daten direkt aus der XML-Datei lesen. Danach werden die Umwelt, die Agenten und Moskitos erstellt. Während des Simulationslaufes werden die Positionen und Zustände der Agenten und Moskitos auf der grafischen Benutzeroberfläche aktualisiert. Bei der Konsolenausgabe findet der selbe Ablauf, wie bei der grafischen Benutzeroberfläche statt, nur das keine grafische Ausgabe erfolgt, stattdessen werden Kommentare auf der Konsole ausgegeben, u.a. Start Simulation, End Simulation. Das Konfigurationstool liest und schreibt nur die XML-Datei. Die Teilabläufe vom Agent-Prozess und Moskito-Prozess werden im Folgenden erläutert.

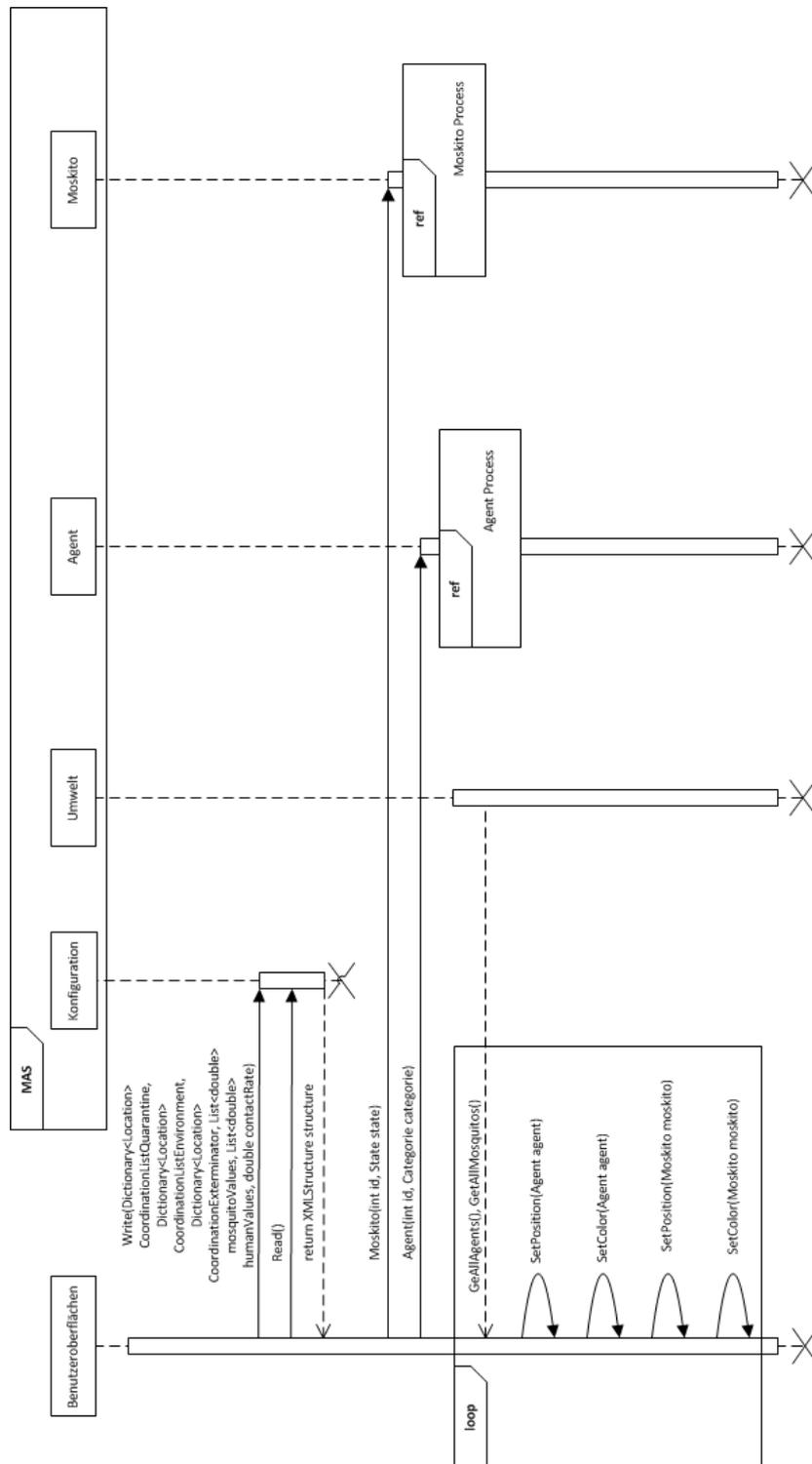


Abbildung 5.2: Ablaufdiagramm des Prototypen

In Abbildung 5.3 ist die abgekapselte Laufzeitsicht des Agenten zu sehen.

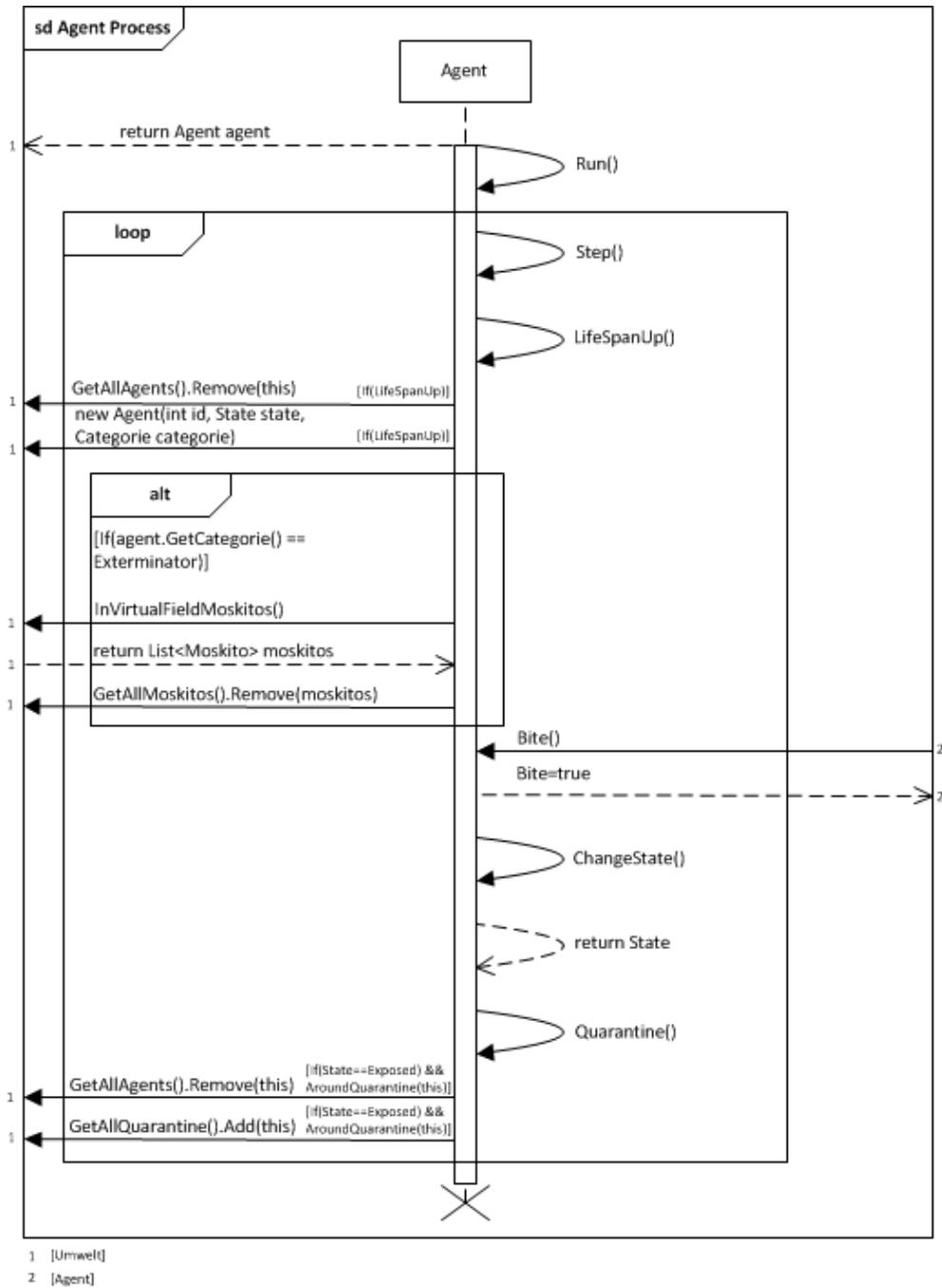


Abbildung 5.3: Ablaufdiagramm des Prototypen - Agent Process

Der Agent wird durch die *Run()* Methode gestartet. In regelmäßigen Abständen wird eine neue Position (*Step()*) berechnet, die als zukünftige Position gesetzt wird, zusätzlich erfolgt die Änderung des Zustands (*ChangeState()*) nach einem Stich (*Bite()*) eines infizierten Moskitos. Zudem erfolgt die Abfrage des Zustands des Agenten, ist dieser Zustand Infiziert oder Infektiös (*[If((State==Exposed)||(State==Infectious)) [...]]*) und der Agent befindet sich in der Nähe einer Quarantänestation (*[...] AroundQuarantine(this)]*), wird der Agent in die Quarantänestation eingewiesen (*GetAllQuarantine().Add(this)*), dort für eine bestimmte Zeit behandelt und der Agent wird aus der Agentenliste entfernt (*GetAllAgents().Remove(this)*). Dies schützt den Agenten in der Quarantänestation vor äußeren Einflüssen. Nachdem der Agent aus der Quarantänestation entlassen wurde, wird der Agent der Agentenliste wieder hinzugefügt. Außerdem spielt die Lebensdauer (*LifeSpanUp()*) von Agenten eine Rolle, die auch in regelmäßigen Abständen überprüft wird. Ist die Lebensdauer abgelaufen, wird der Agent gelöscht (*GetAllAgents().Remove(this)*) und es wird ein neuer Agent (*new Agent(int id, State state, Kategorie categorie)*) in dem Zustand (Anfällig oder Infiziert) des verstorbenen Agenten erstellt. Ist der Agent der Kategorie Exterminator (Schädlingsbekämpfer) zugeordnet, gibt es einen zusätzlichen Ablauf. In regelmäßigen Abständen fragt der Schädlingsbekämpfer nach einem virtuellen Feld (*InVirtualFieldMoskito()*), in dem sich Moskitos befinden. Diese Moskitos werden dann gelöscht (*GetAllMoskitos().Remove(Moskitos)*).

In Abbildung 5.4 ist die abgekapselte Laufzeitsicht des Moskito zu sehen.

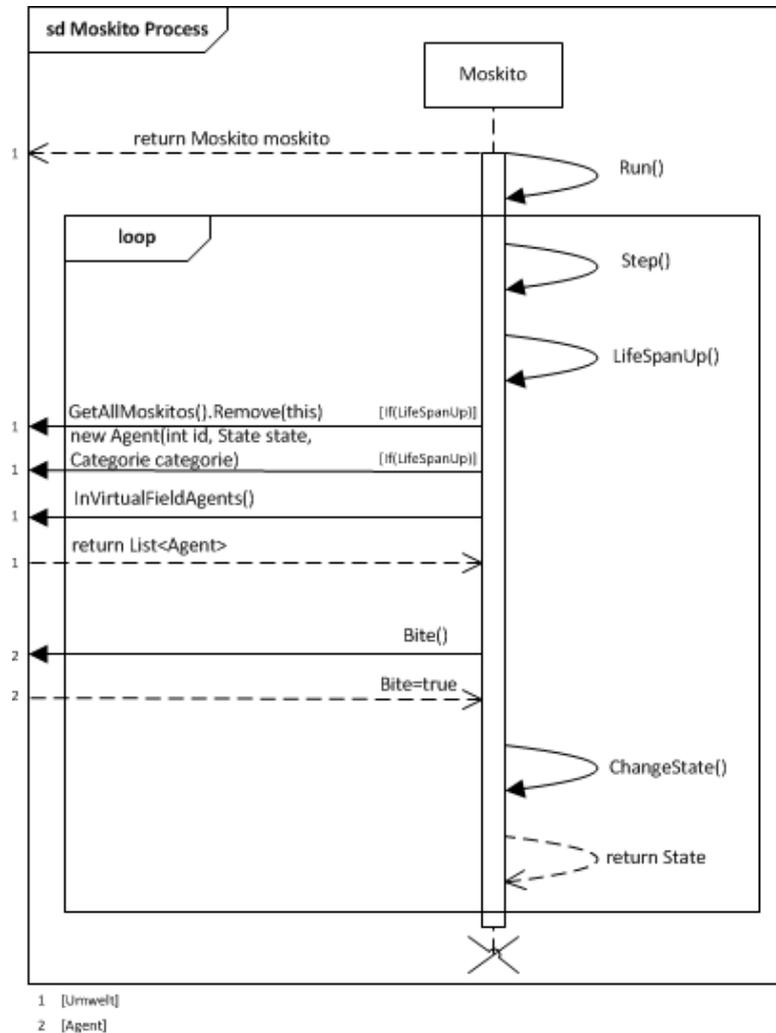


Abbildung 5.4: Ablaufdiagramm des Prototypen - Moskito Process

Der Moskito wird durch die *Run()* Methode gestartet. In regelmäßigen Abständen wird eine neue Position (*Step()*) berechnet, die als zukünftige Position gesetzt wird, zusätzlich erfolgt die Änderung des Zustands (*ChangeState()*), wenn der Moskito infiziert ist. Außerdem spielt die Lebensdauer (*LifeSpanUp()*) des Moskito eine Rolle, die auch in regelmäßigen Abständen überprüft wird. Ist die Lebensdauer abgelaufen, wird der Moskito gelöscht (*GetAllMoskitos().Remove(this)*) und es wird ein neuer Moskito (*new Moskito(int id, State state)*) in dem Zustand (Anfällig oder Infiziert) des verstorbenen Moskito erstellt. In regelmäßigen Abständen fragt der Moskito sein virtuelles Feld (*InVirtualFieldAgents()*) ab. Aus dieser Liste wird zufällig ein Agent ausgewählt und gebissen.

## 6 Realisierung

Die nächsten Abschnitte beschreiben die Realisierungsumgebung, sowie die Realisierung des Anwendungskerns mit seinen Komponenten, der Fassade, die Benutzeroberflächen, dem Adapter und das Dateisystem.

### 6.1 Realisierungsumgebung

Die Tabelle 6.1 zeigt die benutzte Umgebung zur Realisierung.

<b>.NET Framework</b>	.NET Framework 4.5
<b>Entwicklungsumgebung</b>	Microsoft Visual Studio Ultimate 2012
<b>Oberflächengestaltung</b>	Windows Presentation Foundation (WPF)
<b>Betriebssystem</b>	Microsoft Windows 7 Professional
<b>Hilfsmittel</b>	ReSharper (R#) Microsoft.Maps.MapControl.WPF.dll

Tabelle 6.1: Realisierungsumgebung - Eigenschaften

Die Komponenten des Anwendungskerns und die Benutzeroberflächen (Konsole, Grafische und KonfigurationsTool) des Prototypen wurden mit dem *.NET Framework 4.5* realisiert. Das Framework wird durch die Entwicklungsumgebung *Microsoft Visual Studio Ultimate 2012* bereitgestellt. Zusätzlich wurde die grafische Benutzeroberfläche mit dem Grafik Framework *Windows Presentation Foundation (WPF)* erweitert. Die Entwicklung wurde auf dem *Microsoft Windows 7 Professional* Betriebssystem durchgeführt. Weitere Hilfsmittel, die die Entwicklung vereinfachten, waren ReSharper (R#) und das Kartenmaterial von *bing (Microsoft.Maps.MapControl.WPF.dll)* (Microsoft (2014)). ReSharper (R#) ist ein ergänzender Softwarebaustein von JetBrains (JetBrains (2014)), der die Code-Analyse parallel zum Programmieren ausführt und dadurch Verbesserungsvorschläge anbietet, die übernommen werden können, aber nicht müssen.

## 6.2 Anwendungskern

Die nächsten Abschnitte erläutern den Anwendungskern, dessen einzelne Komponenten Umwelt, Agent, Moskito und Konfiguration

### Umwelt

Die Komponente Umwelt hält die Agenten und Moskitos (siehe Anforderung **U1**). Zudem hält die Komponente die Koordinaten der Umwelt und der Quarantänestationen (siehe Anforderung **U2** und **U3**). Außerdem wird das Epidemiemodell in der Komponente gehalten. Beim Erstellen einer Instanz von Umwelt werden die maximalen und minimalen x- und y-Koordinaten berechnet, die später für die Berechnung einer zufälligen Position eines Agenten bzw. Moskito (siehe Anforderung **U5**) wichtig sind. Die erste Position wird zufällig berechnet und bei der Erstellung der Agenten und Moskitos benötigt. Die nächste Position wird aus der vorherigen Position und einer Richtung (rechts, links, oben und unten) ermittelt. Das Epidemiemodell hält die Anzahl der Agenten je Zustand in Abhängigkeit von der Zeit (siehe Anforderung **U6**) fest.

Der nächste Abschnitt zeigt, wie ermittelt wird, ob sich die Position in der Umwelt<sup>1</sup>/Nähe einer Quarantänestation bzw. VirtualField<sup>2</sup> befindet, allgemein wird dies auch als Punkt in einem Polynom bezeichnet (Vgl. **Orwant u. a. (2000)**).

---

<sup>1</sup>Der Lebensraum von Agent und Moskito

<sup>2</sup>Enthält alle Agenten, die sich in der Nähe eines Moskitos/ einer Quarantänestation befinden

### Punkt in einem Polygon

Die Abbildung 6.1 zeigt ein Polygon mit acht Punkten, die sich innerhalb und außerhalb des Polygons befinden.

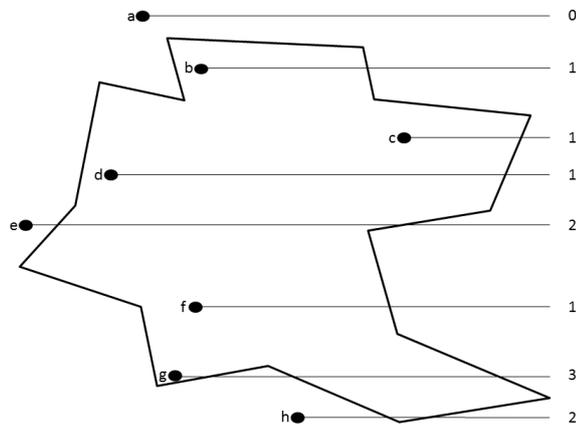


Abbildung 6.1: Punkt in einem Polygon

(In Anlehnung an: [Orwant u. a. \(2000\)](#))

Jeder Punkt bekommt einen Strahl, der außerhalb des Polygons endet, zugeordnet. Der Algorithmus gibt an, ob sich ein Punkt in einem Polygon befindet oder nicht, dies erfolgt durch die Zählung der Überschneidungen des Strahls mit dem Polygon. Wenn die Anzahl ungerade (b, c, d, f, g) ist, liegt der Punkt innerhalb des Polygons, bei einer geraden (a, e, h) Anzahl befindet der Punkt sich nicht im Polygon.

Im Anhang [A.1 \(C# Code - InEnvironment\(\) und InVirtualField\(\)\)](#) ist der Code zu sehen, der in den Methoden `InEnvironment(Location location)`, `AroundQuarantine(Agent agent)`, `InVirtualFieldAgents(Location location)` und `InVirtualFieldMoskitos(Location location)` realisiert wurde. Die Methoden `InEnvironment(Location location)` und `AroundQuarantine(Agent agent)` geben einen booleschen Wert zurück der angibt, ob sich der Agent bzw. Moskito in der Umwelt bzw. Quarantänestation befindet. Bei den Methoden `InVirtualFieldAgents(Location location)` und `InVirtualFieldMoskitos(Location location)` wird jeweils eine Liste zurückgegeben, die Agenten bzw. Moskitos enthält, die sich im virtuellen Feld des Moskitos bzw. Agenten befinden (siehe Anforderung [U4](#)).

### Agent

In der Komponente Agent wird ein Agent mit einer Id und einer Kategorie (siehe Anforderung A1) erstellt. Zudem werden noch folgende Eigenschaften gesetzt: der Zustand in *Anfällig*, die aktuelle und zukünftige Position und die Eigenschaft, ob der Agent gebissen worden ist oder nicht. Nachdem der Agent erfolgreich erstellt wurde, kann sich der Agent frei in der Umwelt bewegen (siehe Anforderung A6). Nach jedem Schritt wird eine neue Position ermittelt, die aktuelle Position durch die zukünftige Position ersetzt und die zukünftige Position wird mit der neu ermittelten Position belegt. Nachdem der Agent gestartet wurde, werden in regelmäßigen Abständen die Lebensdauer, Zustandsänderungen und Quarantänestationen abgefragt. Ist die Lebensdauer abgelaufen, wird der Agent gelöscht und ein neuer Agent erstellt (siehe Anforderung A7). Die Zustandsänderung erfolgt erst nach einem Moskitobiss bzw. wenn sich der Agent nach der Geburt im Zustand *Infiziert* befindet. Wird der Agent von einem infektiösen Moskito gebissen, wechselt der Agent vom Zustand *Anfällig* in den Zustand *Infiziert* (siehe Anforderung A2). Die weiteren Zustandsübergänge erfolgen nach einer Berechnung. Ist der berechnete Wert kleiner oder gleich den definierten Raten (u.a. Übertragungsrate, Infektionsrate), geht der Agent in den nächsten Zustand, in diesem Fall *Infektiös* oder *Geheilt* (siehe Anforderung A3 und A4), über. Ist der berechnete Wert größer, bleibt der Agent im aktuellen Zustand. Zusätzlich wird die natürliche und von der Krankheit verursachte Sterberate (siehe Anforderung A7) berücksichtigt. Wird die Eigenschaft Wiederinfizierung gesetzt, wird der Agent vom Zustand *Infektiös* in den Zustand *Anfällig* gesetzt und die Eigenschaft, ob der Agent gebissen wurde, zurückgesetzt (siehe Anforderung A5). Befindet sich der Agent im Zustand *Infiziert* oder *Infektiös* und in der Nähe einer Quarantänestation wird der Agent in der Quarantänestation aufgenommen und verweilt dort. Durch die Aufnahme in eine Quarantänestation wird der Agent von der Außenwelt abgeschottet und hat eine Chance zu überleben (siehe Anforderung A8). Nachdem sich der Agent im Zustand *Geheilt* befindet, wird er aus der Quarantäne Station entlassen. Ist der Agent der Kategorie *Schädlingsbekämpfer* zugeordnet und befindet sich in keiner Quarantänestation, wird dieser in regelmäßigen Abständen aktiv, indem der Schädlingsbekämpfer alle in seinem virtuellen Feld befindlichen Moskitos beseitigt (siehe Anforderung A9). Zum Programmstart wird die Reproduktionsnummer berechnet, die angibt, ob es sich um eine Epidemie handelt (siehe Anforderung A10).

### Moskito

In der Komponente Moskito wird ein Moskito mit einer Id bzw. einer Id und einem Zustand (siehe Anforderung M1 und M2) erstellt. Zudem werden noch folgende Eigenschaften gesetzt:

die aktuelle und zukünftige Position. Nachdem der Moskito erfolgreich erstellt worden ist, kann der Moskito sich frei in der Umwelt bewegen (siehe Anforderung M5). Nach jedem Schritt wird eine neue Position ermittelt. Wie bei der Komponente Agent, wird die aktuelle Position mit der zukünftigen Position und die zukünftige Position mit der neu ermittelten Position belegt. Nachdem der Moskito gestartet wurde, wird in regelmäßigen Abständen die Lebensdauer und eine Zustandsänderung abgefragt. Ist die Lebensdauer abgelaufen, wird der Moskito gelöscht und ein neuer Moskito im Zustand des verstorbenen Moskitos erstellt (siehe Anforderung M6). Der Moskito lässt sich in regelmäßigen Abständen sein virtuelles Feld geben und beißt einen Agenten aus dem virtuellen Feld (siehe Anforderung M7). Eine initiale Zustandsänderung erfolgt nach einem Biss, wenn sich der gebissene Agent sich im Zustand *Infektiös* und der beissende Moskito im Zustand *Anfällig* befinden (siehe Anforderung M3). Ein weiterer Zustandsübergang erfolgt nach einer Berechnung eines Wertes. Ist der berechnete Wert kleiner oder gleich der Übertragungsrate geht der Moskito in den nächsten Zustand, in diesem Fall *Infektiös* (siehe Anforderung M4), über. Der Moskito bleibt sein Leben lang im Zustand *Infektiös* (siehe Anforderung M2). Ist der berechnete Wert größer, bleibt der Moskito im aktuellen Zustand. Weiterhin wird noch die natürliche Sterberate (siehe Anforderung M6), wie bei der Zustandsänderung, mit einer Berechnung berücksichtigt.

### Konfiguration

Die Komponente Konfiguration ist dafür zuständig die Eigenschaftswerte aus der XML-Datei auszulesen und in die Datei zuschreiben. Folgende Eigenschaften werden in der Konfiguration gehalten: Kontaktrate (siehe Anforderung K20), Sterberate vom Agent und Moskito (siehe Anforderung K15), Krankheitsbedingte Sterberate des Agenten (siehe Anforderung K17), Lebensdauer vom Agent und Moskito (siehe Anforderung K13), Nachkommensgröße vom Agent und Moskito (siehe Anforderung K14), Geburtenrate vom Agent (siehe Anforderung K18), Übertragungsrate vom Agent und Moskito (siehe Anforderung K16), Infektionsrate vom Agent (siehe Anforderung K19), Populationsgröße im Zustand *Anfällig* vom Agent und Moskito (siehe Anforderung K11 und K12), Populationsgröße im Zustand *Infektiös* vom Moskito (siehe Anforderung K12), Koordinaten der Umwelt und Quarantänestationen (siehe Anforderung K21) und die Wiederinfizierung vom Agent (siehe Anforderung K22).

### 6.3 Fassade

Die Fassade stellt folgende Methoden bereit:

- a) Factorymethoden zum Erzeugen von Moskitos und Agenten

- b) Zugriff auf die Methode *Run()* vom Mosquito und Agent
- c) Zugriff auf die Methode *SetProperties()* und *GetProperties()* für das Schreiben und Lesen von Eigenschaftswerten
- d) Zugriff auf die Methode *Write* für das Schreiben der Werte des Epidemiemodells in eine Datei

### 6.4 Benutzeroberflächen

Die nächsten Abschnitte beschreiben die grafische Benutzeroberfläche, die Konsolenausgabe und das KonfigurationsTool.

#### Grafische Benutzeroberfläche

Die Abbildung 6.2 zeigt die Benutzeroberfläche vor dem Start einer Simulation.



Abbildung 6.2: Vor dem Start einer Simulation

Nachdem die grafische Benutzeroberfläche gestartet wurde, werden die Daten aus der XML-Datei geladen und auf der grafischen Benutzeroberfläche dargestellt, u.a. Krankenhäuser, die Grenzen der Umwelt und die Eigenschaften für die Simulation. Vor dem eigentlichen Start der Simulation können diese Daten über den Button *Properties* verändert werden. Es wird das KonfigurationsTool, das im übernächsten Abschnitt **KonfigurationsTool** erläutert wird, aufgerufen, dass die Möglichkeit bietet, diese Daten einzustellen bzw. zu ändern.

## 6 Realisierung

Nach Änderung von Daten für die Simulation werden diese geänderten Werte auf der grafischen Benutzeroberfläche angezeigt, wie es in der Abbildung 6.3 zu sehen ist.

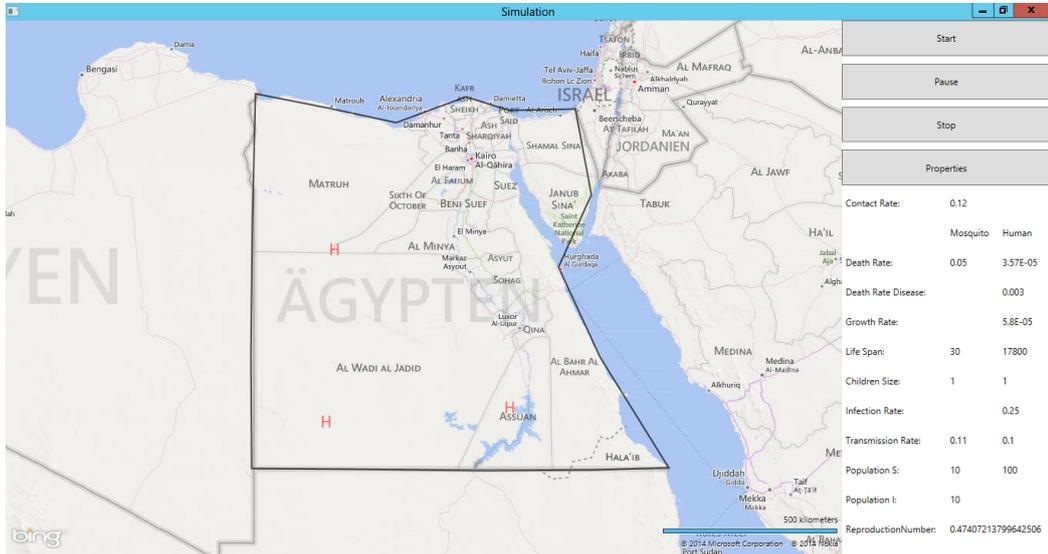


Abbildung 6.3: Nach den Einstellen der Daten für die Simulation

Zur Abbildung 6.2 haben sich einige Daten verändert, u.a. Standorte der Quarantänestationen, Eigenschaften der Simulation und die Grenzen der Umwelt, wie in der Abbildung 6.3 zu sehen ist.

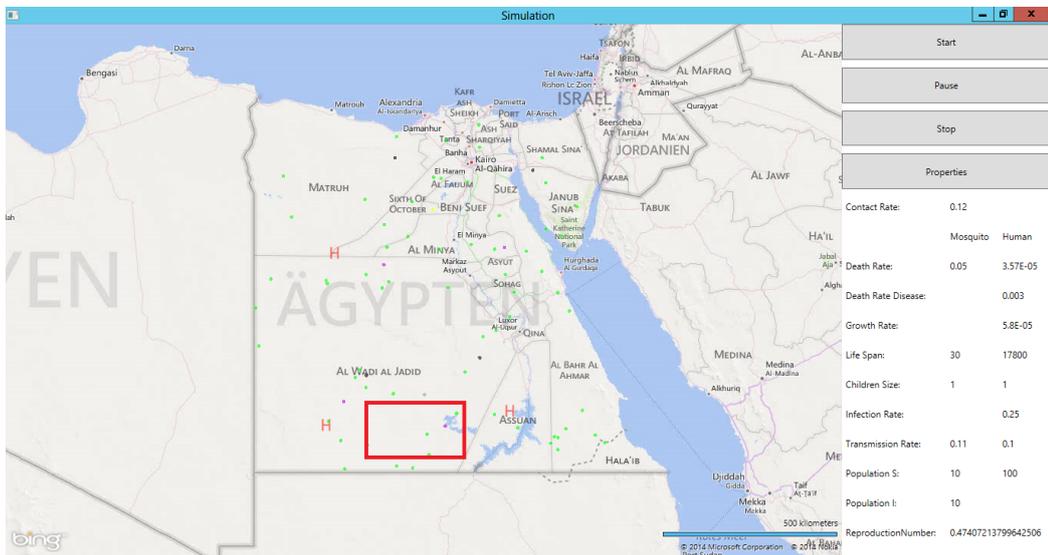


Abbildung 6.4: Simulationsdurchlauf

Es ist möglich die Simulation anzuhalten (pausieren) und fortzusetzen, sowie die Simulation zu beenden. Die Abbildung 6.4 zeigt die Agenten in verschiedenen Farben, je nachdem in welchen Zustand die Agenten sich befinden. Die Farbe grün für den Zustand *Anfällig*, die Farbe gelb für *Infiziert*, die Farbe rot für *Infektiös* und die Farbe grau für *Geheilt*. Der Moskito wird in der Farbe schwarz für den Zustand *Anfällig*, in der Farbe blau für *Infiziert* und in der Farbe violett für *Infektiös* dargestellt.

Die Abbildung 6.5 zeigt ein virtuelles Feld eines Moskitos, indem sich Agenten befinden, deren Risiko, gebissen zu werden, hoch ist.

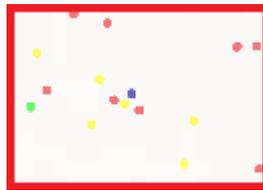


Abbildung 6.5: Virtuelles Feld eines Moskitos

Das virtuelle Feld des Moskitos, das in Abbildung 6.5 zu sehen ist, zeigt die Ausbreitung einer Infektionskrankheit nach dem SEIR-Epidemiemodell. Je nachdem, ob sich ein oder mehrere Agenten im virtuellen Feld des Moskitos befindet bzw. befinden, sucht sich der Moskito zufällig einen Agenten aus und beißt ihn. Trägt der Moskito die Krankheit in sich, wird der Agent infiziert. Nach dem Biss erfolgt der Durchlauf des SEIR-Epidemiemodells. Der Agent wird in Abhängigkeit seines Zustandes verschiedenfarbig dargestellt. Die Dauer des Durchlaufs des SEIR-Epidemiemodells wird durch die Werte für Inkubations- und Infektionszeit bestimmt.

### **Konsolenausgabe**

Die Konsolenausgabe stellt eine Simulation ohne Benutzeroberfläche dar. Während der Ausführung erfolgt nur eine Ausgabe von Kommentaren u.a. Start Simulation, End Simulation. Am Ende der Simulation wird eine Datei geschrieben, die die SEIR-Epidemiekurve der Agenten enthält.

## KonfigurationsTool

Das KonfigurationsTool ist dafür zuständig, Voreinstellungen für die Simulation zu definieren. Die Abbildung 6.6 zeigt das KonfigurationsTool im Initialzustand ohne gesetzte Parameter.

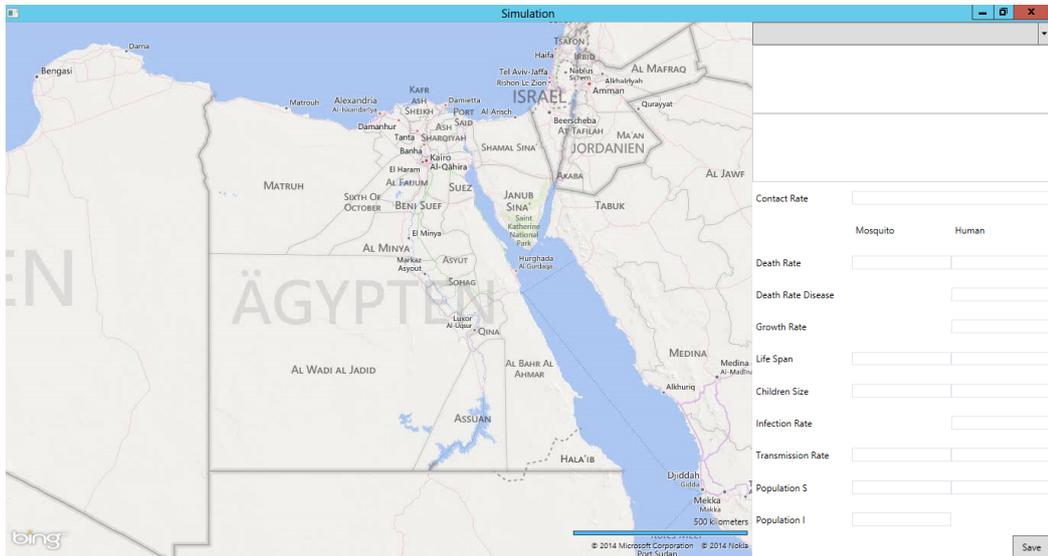


Abbildung 6.6: KonfigurationsTool im Initialzustand ohne gesetzte Parameter

Um einen Simulationsdurchlauf durchführen zu können, müssen einige Voreinstellungen vorgenommen werden. Die Voreinstellungen beziehen sich auf die Parameter Kontaktrate, Nachkommensgröße, Sterberate, Sterberate bezüglich der Krankheit, Geburtenrate, Infektionsrate, Übertragungsrate, Lebensdauer und die Größe der Population. Die Parameter sind in der Abbildung 6.7 unten rechts zu sehen. Ergänzende Voreinstellungen sind noch nötig: Bereichs- bzw. Ländergrenzen für die Umwelt und Standorte von Quarantänestationen. Jede Position wird durch einen Punkt repräsentiert, der in der Abbildung zu sehen sind. Die Punkte, die in die Karte (Map) eingezeichnet werden, sind oben rechts als Liste geführt, unterschieden in den Kategorien Umwelt und Quarantänestation.

Die Abbildung 6.7 zeigt das KonfigurationsTool nachdem Simulationsparameter eingestellt wurden.

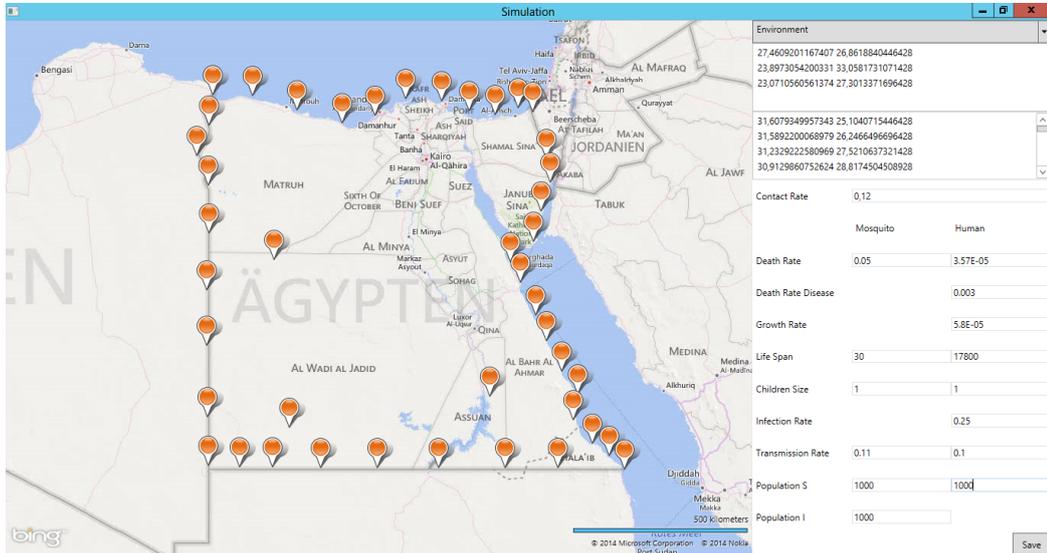


Abbildung 6.7: KonfigurationsTool nachdem Simulationsparameter eingestellt wurden

Das .NET Framework bietet den XmlSerializer an, der genutzt wurde, um XML-Dateien zu schreiben und wieder auszulesen. Die XML-Datei wurde mit Hilfe einer XML-Struktur beschrieben. Im Anhang A.2 (XML-Struktur der XML-Datei) ist die XML-Struktur der XML-Datei zu sehen.

## 6.5 Adapter

Der Adapter erstellt eine Datei mit der gegebenen XML-Struktur.

## 6.6 Dateisystem

Die Datei im XML-Format wird direkt auf der Festplatte gespeichert. Wird die Datei benötigt, wird sie direkt von der Festplatte gelesen.

## 7 Ergebnisse und Diskussion

In diesem Abschnitt werden die Simulationsergebnisse des Prototyps dargestellt und diskutiert.

### 7.1 Simulationsumgebung

Die Tabelle 7.1 zeigt die Ausstattung des Simulationsrechners, der für die Simulationsdurchläufe benutzt wurde.

<b>CPU</b>	QEMU Virtual CPU version 2.0.0 2.79 GHz (12 Prozessoren)
<b>RAM</b>	48,0 GB
<b>Betriebssystem</b>	Windows Server 2012 Datacenter, 64 Bit
<b>.NET Framework</b>	.NET Framework 4.5

Tabelle 7.1: Simulationsumgebung - Eigenschaften

### 7.2 Tropische Infektionskrankheiten - Parameter

Die Tabelle 7.2 zeigt die Default-Werte, die für alle vier Infektionskrankheiten verwendet wird.

<b>Variable</b>	<b>Wert</b>	<b>Beschreibung</b>
$\lambda_h$	0,000058	Geburtenrate
$\mu_h$	0,000035	Natürliche Todesrate
$S_h$	10.000	Anzahl an anfälligen Agenten
$S_v$	100.000	Anzahl an anfälligen Moskitos
$I_v$	10.000	Anzahl an infektiösen Moskitos
$E_v, E_h, I_h, R_h$	0	Anzahl an infizierten Moskitos, infizierten, infektiösen und geheilten Agenten
t	200	Anzahl der Simulationsschritte

Tabelle 7.2: Allgemein - Parameter

Die Geburtenrate ( $\lambda_h$ ) beläuft sich auf  $5.8 * 10^{-5}$  pro 1000/Jahr und die natürliche Todesrate ( $\mu_h$ ) auf  $3.5 * 10^{-5}$  pro 1000/Jahr. Die Einheit 1000/Jahr beschreibt die Anzahl der Geburten bzw. Todesfälle pro Jahr bezogen auf 1000 Einwohner. Die weiteren Parameter beschreiben die Anzahl der Population, je nach Zustand und Art. Die Anzahl der Agenten  $S_h$  im Zustand *Anfällig* beläuft sich auf 10.000 Individuen, die Anzahl der Moskitos im Zustand *Anfällig*  $S_v$  beläuft sich auf 100.000 Moskitos und im Zustand *Infektiös*  $I_v$  auf 10.000 Moskitos. Die Anzahl der Moskitos im Zustand *Infiziert*  $E_v$  und die Agenten im Zustand *Infektiös*  $E_h$ , *Infektiös*  $I_h$  und *Geheilt*  $R_h$  sind auf 0 definiert (Vgl. Erickson u. a. (2010)). Die Zeiteinheit (t) beläuft sich auf 200 Simulationsschritte, die Schritte resultieren aus den durchschnittlichen Simulationsdurchläufen, in einer Simulation. Die Erläuterung der einzelnen Parameter sind im Abschnitt 3.4 (**Vector-born Epidemie Modell**) zu finden.

### Dengue-Fieber

In der Tabelle 7.3 sind die Parameterwerte der tropischen Infektionskrankheit Dengue-Fieber zu finden, die für die Simulation verwendet werden.

Variable	Wert	Beschreibung
$\tau e_h$	0,1	Übertragungsrate <sup>a</sup> der Agenten
$\tau i_h$	0,25	Infektionsrate der Agenten
$\alpha_h$	0,003	Durch die Infektionskrankheit verursachte Todesrate bei Agenten
$\mu_v$	0,05	Natürliche Todesrate der Moskitos
$\tau e_v$	0,11	Übertragungsrate <sup>a</sup> der Moskitos
c	0,12	Kontaktrate

Tabelle 7.3: Dengue-Fieber - Parameter

Die Übertragungsrate von Agenten ( $\tau e_h$ ) beläuft sich auf 0,1, die Infektionsrate der Agenten ( $\tau i_h$ ) auf 0,25, die durch die tropische Infektionskrankheit verursachte Todesrate bei Agenten ( $\alpha_h$ ) auf 0,003, die natürliche Todesrate von Moskitos ( $\mu_v$ ) auf 0,05, die Übertragungsrate von Moskitos ( $\tau e_v$ ) auf 0,11 und die Kontaktrate (c) auf 0,12 (Vgl. Erickson u. a. (2010)).

<sup>a</sup>Wahrscheinlichkeit des Übergangs vom Zustand Anfällig in den Zustand Infiziert

## Malaria

In der Tabelle 7.4 sind die Parameterwerte der tropischen Infektionskrankheit Malaria zu finden, die für die Simulation verwendet werden.

Variable	Wert	Beschreibung
$\tau e_h$	0,034	Übertragungsrate der Agenten
$\tau i_h$	0,00136	Infektionsrate der Agenten
$\alpha_h$	0,05	Durch die Infektionskrankheit verursachte Todesrate bei Agenten
$\mu_v$	0,10483	Natürliche Todesrate der Moskitos
$\tau e_v$	0,09	Übertragungsrate der Moskitos
c	0,05	Kontaktrate

Tabelle 7.4: Malaria - Parameter

Die Übertragungsrate von Agenten ( $\tau e_h$ ) beläuft sich auf 0,034, die Infektionsrate der Agenten ( $\tau i_h$ ) auf 0,00136, die durch die tropische Infektionskrankheit verursachte Todesrate bei Agenten ( $\alpha_h$ ) auf 0,05, die natürliche Todesrate ( $\mu_v$ ) von Moskitos auf 0,10483, die Übertragungsrate ( $\tau e_v$ ) von Moskitos auf 0,09 und die Kontaktrate (c) auf 0,05 (Vgl. [Erickson u. a. \(2010\)](#)).

## West-Nile-Fieber

In der Tabelle 7.5 sind die Parameterwerte der tropischen Infektionskrankheit West-Nile-Fieber zu finden, die für die Simulation verwendet werden.

Variable	Wert	Beschreibung
$\tau e_h$	0,25	Übertragungsrate der Agenten
$\tau i_h$	0,5	Infektionsrate der Agenten
$\alpha_h$	0,004	Durch die Infektionskrankheit verursachte Todesrate bei Agenten
$\mu_v$	0,01457	Natürliche Todesrate der Moskitos
$\tau e_v$	0,1847	Übertragungsrate der Moskitos
c	0,344	Kontaktrate

Tabelle 7.5: West-Nile-Fieber - Parameter

Die Übertragungsrate von Agenten ( $\tau e_h$ ) beläuft sich auf 0,25, die Infektionsrate der Agenten ( $\tau i_h$ ) auf 0,5, die durch die tropische Infektionskrankheit verursachte Todesrate bei Agenten

( $\alpha_h$ ) auf 0,004, die natürliche Todesrate ( $\mu_v$ ) von Moskitos auf 0,01457, die Übertragungsrate ( $\tau e_v$ ) von Moskitos auf 0,1847 und die Kontaktrate (c) auf 0,344 (Vgl. [Erickson u. a. \(2010\)](#)).

### Chikungunya-Fieber

In der Tabelle 7.6 sind die Parameterwerte der tropischen Infektionskrankheit Chikungunya-Fieber zu finden, die für die Simulation verwendet werden.

Variable	Wert	Beschreibung
$\tau e_h$	0,22	Übertragungsrate der Agenten
$\tau i_h$	0,13	Infektionsrate der Agenten
$\alpha_h$	0	Durch die Infektionskrankheit verursachte Todesrate bei Agenten
$\mu_v$	0,1	Natürliche Todesrate der Moskitos
$\tau e_v$	0,5	Übertragungsrate der Moskitos
c	0,6	Kontaktrate

Tabelle 7.6: Chikungunya-Fieber - Parameter

Die Übertragungsrate von Agenten ( $\tau e_h$ ) beläuft sich auf 0,22, die Infektionsrate der Agenten ( $\tau i_h$ ) auf 0,13, die durch die tropische Infektionskrankheit verursachte Todesrate bei Agenten ( $\alpha_h$ ) auf 0, d.h. es gibt keine Todesrate, die durch die tropische Infektionskrankheit verursacht wird, die natürliche Todesrate ( $\mu_v$ ) von Moskitos auf 0,1, die Übertragungsrate ( $\tau e_v$ ) von Moskitos auf 0,5 und die Kontaktrate (c) auf 0,6 (Vgl. [Erickson u. a. \(2010\)](#)).

## 7.3 Ergebnisse

Die nächsten Abschnitte zeigen die Ergebnisse von Simulationen in unterschiedliche Szenarien u.a. Default, Quarantänestationen, Schädlingsbekämpfer, Wiederinfizierung und eine Kombination der besten Ergebnisse der Szenarien Quarantänestationen und Schädlingsbekämpfer, sowie eine Zusammenfassung der Ergebnisse nach Szenarien und Krankheiten. Die Simulationszeiten der einzelnen Simulationen fallen unterschiedlich aus. Dies ist darauf zurückzuführen, dass verschiedene Simulationen parallel ausgeführt wurden. Die Anzahl der Prozessoren (siehe Abschnitt 7.1) ist beschränkt, hierdurch wurde die Ausführungszeit einzelner Simulationsdurchläufe beeinträchtigt.

### 7.3.1 Default

Bei dem Szenario Default werden keine Quarantänestationen und Schädlingsbekämpfer verwendet und die Wiederinfizierung wird nicht berücksichtigt.

#### Dengue-Fieber

Die Abbildung 7.1 zeigt das Ergebnis eines Simulationsdurchlaufs der tropischen Infektionskrankheit Dengue-Fieber. Der Simulationsdurchlauf dauerte 50 Minuten und 12 Sekunden.

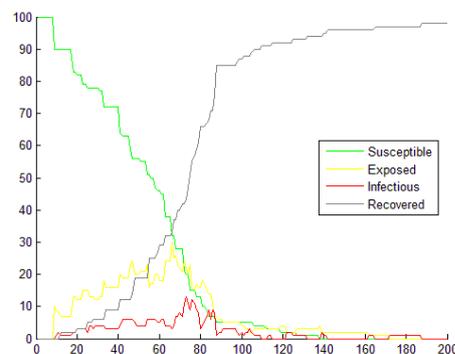


Abbildung 7.1: Default - Dengue-Fieber

Der Epidemie Peak<sup>1</sup> ist am Tag 72. An diesem Tag sind 12% der Human Population infiziert. Die Reproduktionsnummer beläuft sich auf 0,47, d.h. es handelt sich um keine Epidemie, da der Wert unter 1 liegt.

---

<sup>1</sup>Der Höchststand an infektiösen Agenten

## Malaria

Die Abbildung 7.2 zeigt das Ergebnis eines Simulationsdurchlaufs der tropischen Infektionskrankheit Malaria. Der Simulationsdurchlauf dauerte 54 Minuten und 48 Sekunden.

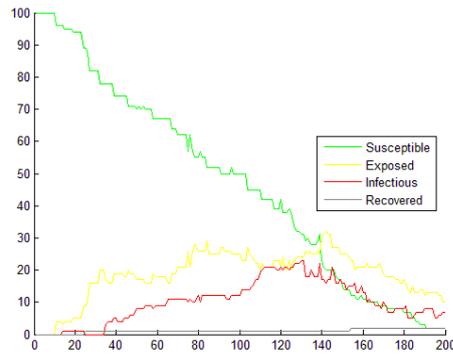


Abbildung 7.2: Default - Malaria

Der Epidemie Peak ist am Tag 130. An diesem Tag sind 22% der Human Population infiziert. Die Reproduktionsnummer beläuft sich auf 0,97, d.h. es handelt sich um keine Epidemie, da der Wert unter 1 liegt.

## West-Nile-Fieber

Die Abbildung 7.3 zeigt das Ergebnis eines Simulationsdurchlaufs der tropischen Infektionskrankheit West-Nile-Fieber. Der Simulationsdurchlauf dauerte 10 Minuten und 38 Sekunden.

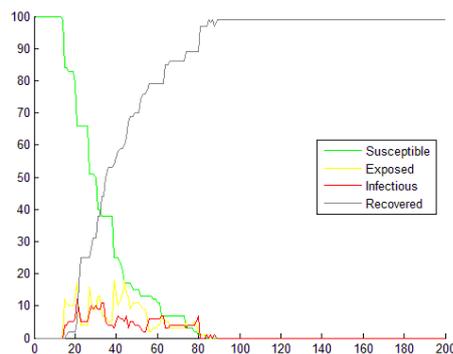


Abbildung 7.3: Default - West-Nile-Fieber

Der Epidemie Peak ist am Tag 21. An diesem Tag sind 12% der Human Population infiziert. Die Reproduktionsnummer beläuft sich auf 0,68, d.h. es handelt sich um keine Epidemie, da der Wert unter 1 liegt.

### Chikungunya-Fieber

Die Abbildung 7.4 zeigt das Ergebnis eines Simulationsdurchlaufs der tropischen Infektionskrankheit Chikungunya-Fieber. Der Simulationsdurchlauf dauerte 6 Minuten und 41 Sekunden.

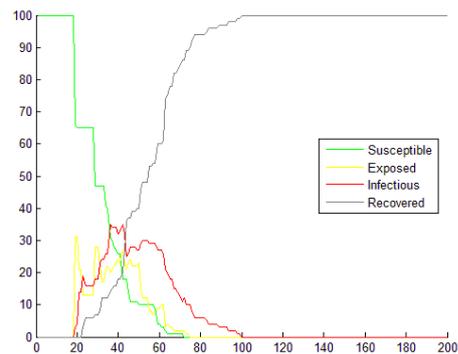


Abbildung 7.4: Default - Chikungunya-Fieber

Der Epidemie Peak liegt zwischen den Tagen 37 und 42. Während dieser Tage sind 34% der Human Population infiziert. Die Reproduktionsnummer kann nicht bestimmt werden, da der Wert der durch die tropischen Infektionskrankheit verursachten Todesrate nicht definiert wurde (Vgl. [Erickson u. a. \(2010\)](#)).

### 7.3.2 Quarantänestationen

Bei dem Szenario Quarantänestationen werden eine, zwei, vier und acht Quarantänestationen definiert. Die Abbildung 7.5 zeigt die Definition einer Quarantänestation. Der Ausschnitt aus der Konfiguration (links) definiert die genauen Koordinaten der Position, die im Bild (rechts) zu sehen ist.

```
1 <CoordinateHospital >  
2 <X>25.404990611478624 </X>  
3 <Y>29.388739513392842 </Y>  
4 </CoordinateHospital >
```



Abbildung 7.5: Position der Quarantänestation - 1 Quarantänestation

Die Abbildung 7.6 zeigt die Definition zweier Quarantänestationen. Der Ausschnitt aus der Konfiguration (links) zeigt die genauen Koordinaten der Positionen, die im Bild (rechts) zu sehen sind.

```
1 <CoordinateHospital >  
2 <X>27.918141655084113 </X>  
3 <Y>29.344794200892842 </Y>  
4 </CoordinateHospital >  
5 <CoordinateHospital >  
6 <X>23.60568775333212 </X>  
7 <Y>29.366766857142842 </Y>  
8 </CoordinateHospital >
```



Abbildung 7.6: Positionen der Quarantänestationen - 2 Quarantänestationen

## 7 Ergebnisse und Diskussion

Die Abbildung 7.7 zeigt die Definition von vier Quarantänestationen. Der Ausschnitt aus der Konfiguration (links) zeigt die genauen Koordinaten der Positionen, die im Bild (rechts) zu sehen sind.

```
1 <CoordinateHospital >
2   <X>29.727573642682096 </X>
3   <Y>29.322821544642842 </Y>
4 </CoordinateHospital >
5 <CoordinateHospital >
6   <X>25.998948561987376 </X>
7   <Y>29.498602794642842 </Y>
8 </CoordinateHospital >
9 <CoordinateHospital >
10  <X>23.545271699023075 </X>
11  <Y>26.949774669642842 </Y>
12 </CoordinateHospital >
13 <CoordinateHospital >
14  <X>23.686199196144813 </X>
15  <Y>32.882391857142842 </Y>
16 </CoordinateHospital >
```

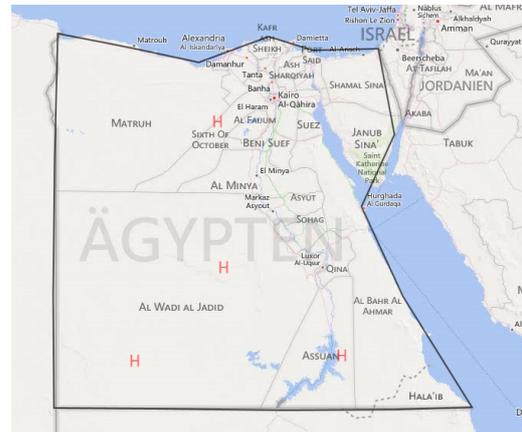


Abbildung 7.7: Positionen der Quarantänestationen - 4 Quarantänestationen

Die Abbildung 7.8 zeigt die Definition der acht Quarantänestationen. Der Ausschnitt aus der Konfiguration (links) zeigt die genauen Koordinaten der Positionen, die im Bild (rechts) zu sehen sind.

```

1 <CoordinateHospital >
2   <X>30.450... </X> <Y>26.378... </Y>
3 </CoordinateHospital >
4 <CoordinateHospital >
5   <X>29.364... </X> <Y>31.564... </Y>
6 </CoordinateHospital >
7 <CoordinateHospital >
8   <X>28.499... </X> <Y>27.565... </Y>
9 </CoordinateHospital >
10 <CoordinateHospital >
11   <X>27.080... </X> <Y>32.025... </Y>
12 </CoordinateHospital >
13 <CoordinateHospital >
14   <X>25.900... </X> <Y>27.125... </Y>
15 </CoordinateHospital >
16 <CoordinateHospital >
17   <X>24.728... </X> <Y>32.025... </Y>
18 </CoordinateHospital >
19 <CoordinateHospital >
20   <X>23.121... </X> <Y>27.323... </Y>
21 </CoordinateHospital >
22 <CoordinateHospital >
23   <X>22.858... </X> <Y>33.915... </Y>
24 </CoordinateHospital >

```

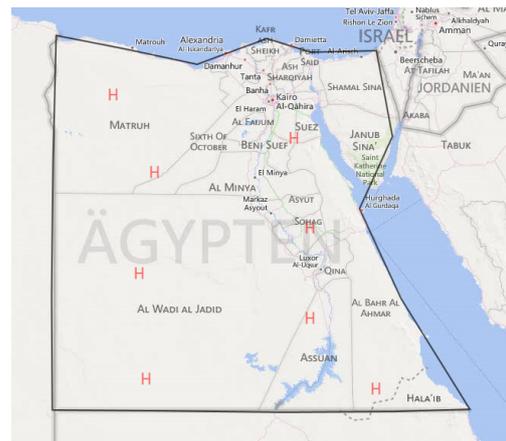


Abbildung 7.8: Positionen der Quarantänestationen - 8 Quarantänestationen

## Dengue-Fieber

Die Abbildungen 7.9 (a-d) zeigen das Ergebnis von vier Simulationsdurchläufen der tropischen Infektionskrankheit Dengue-Fieber. Der erste Simulationsdurchlauf dauerte 40 Minuten und 48 Sekunden bei einer Quarantänestation (a), der Zweite 31 Minuten und 22 Sekunden bei zwei Quarantänestationen (b), der Dritte 42 Minuten und 36 Sekunden bei vier Quarantänestationen (c) und der Letzte 25 Minuten und 2 Sekunden bei acht Quarantänestationen (d).

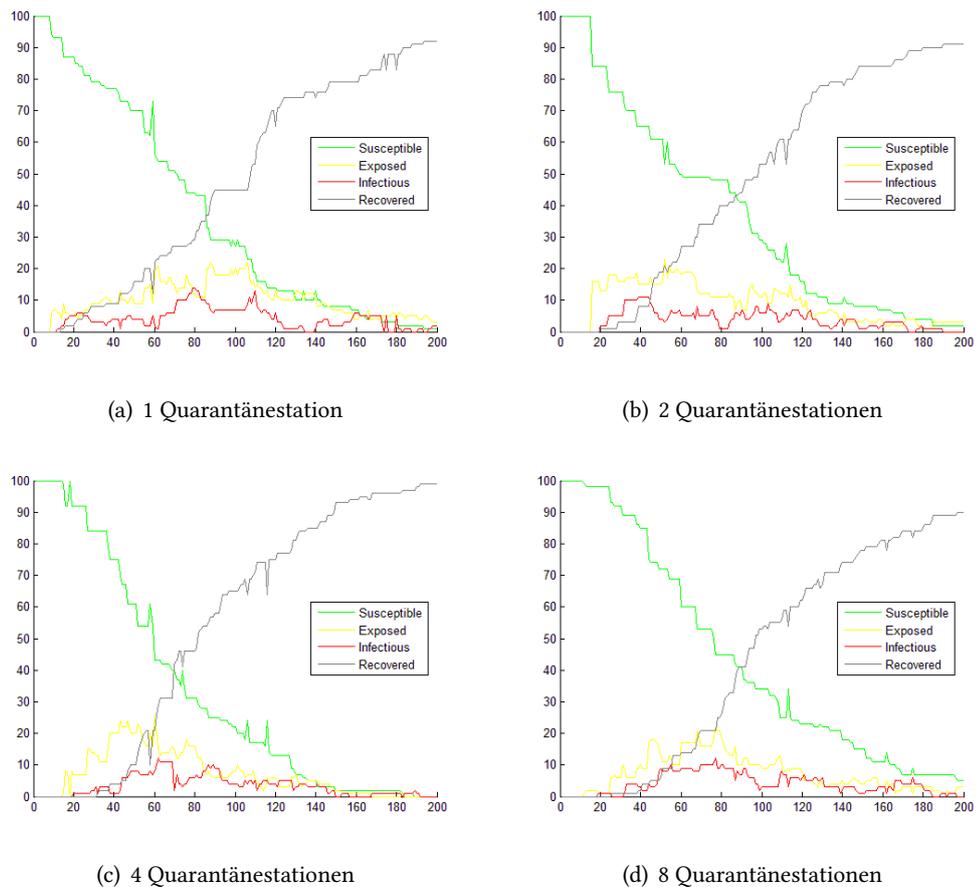


Abbildung 7.9: Unterschiedliche Anzahl an Quarantänestationen - Dengue-Fieber

Der Epidemie Peak bei einer Quarantänestation (a) ist am Tag 79. An diesem Tag sind 14% der Human Population infiziert. Bei zwei Quarantänestationen (b) ist der Epidemie Peak zwischen den Tagen 39-45. In diesem Zeitraum sind etwa 11% der Human Population infiziert. Am Tag 62 ist der Epidemie Peak bei vier Quarantänestationen (c), an diesem Tag sind etwa 13% der Human Population infiziert. Bei acht Quarantänestationen (d) ist der Epidemie Peak am Tag 75 und an diesem Tag sind etwa 13% der Human Population infiziert.

**Malaria**

Die Abbildungen 7.10 (a-d) zeigen das Ergebnis von vier Simulationsdurchläufen der tropischen Infektionskrankheit Malaria. Der erste Simulationsdurchlauf dauerte 32 Minuten und 50 Sekunden bei einer Quarantänestation (a), der Zweite 39 Minuten und 58 Sekunden bei zwei Quarantänestationen (b), der Dritte 28 Minuten und 2 Sekunden bei vier Quarantänestationen (c) und der Letzte 14 Minuten und 34 Sekunden bei acht Quarantänestationen (d).

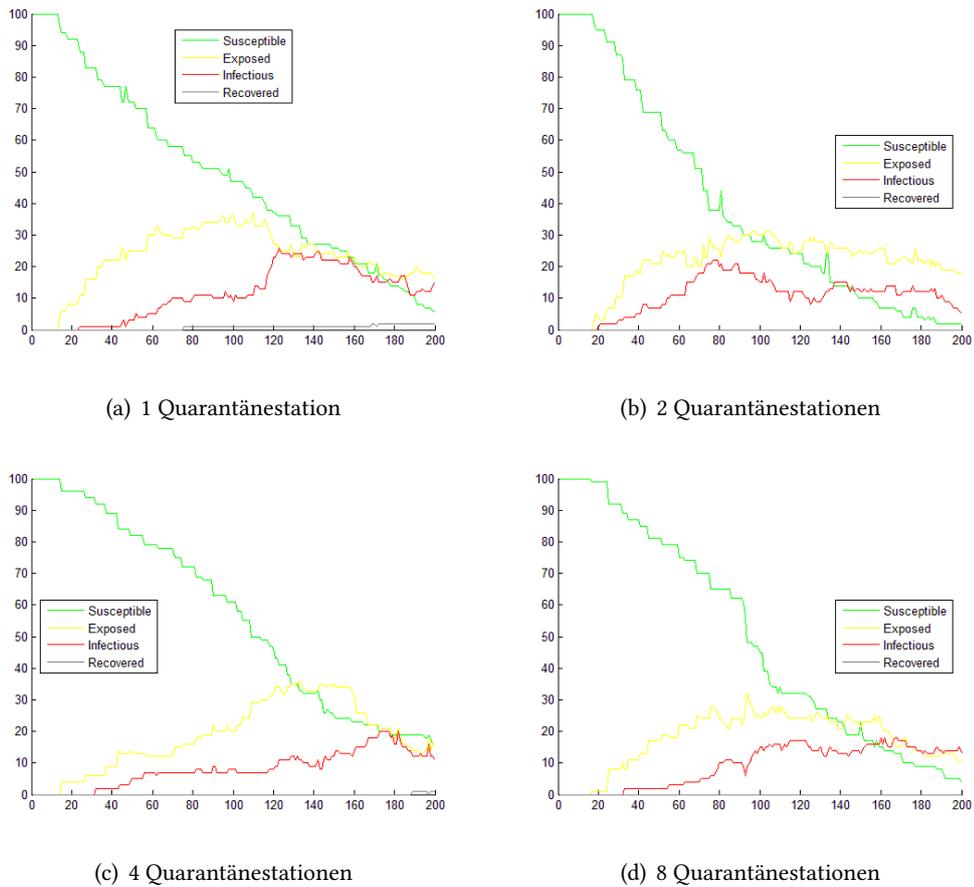


Abbildung 7.10: Unterschiedliche Anzahl an Quarantänestationen - Malaria

Der Epidemie Peak bei einer Quarantänestation (a) ist am Tag 124. An diesem Tag sind 26% der Human Population infiziert. Bei zwei Quarantänestationen (b) ist der Epidemie Peak zwischen den Tagen 78-79. In diesem Zeitraum sind etwa 23% der Human Population infiziert. An den Tagen 173-177 ist der Epidemie Peak bei vier Quarantänestationen (c), an diesen Tagen sind etwa 20% der Human Population infiziert. Bei acht Quarantänestationen (d) ist der Epidemie Peak am Tag 160 und an diesem Tag sind etwa 18% der Human Population infiziert.

### West-Nile-Fieber

Die Abbildungen 7.11 (a-d) zeigen das Ergebnis von vier Simulationsdurchläufen der tropischen Infektionskrankheit West-Nile-Fieber. Der erste Simulationsdurchlauf dauerte 23 Minuten und 22 Sekunden bei einer Quarantänestation (a), der Zweite 12 Minuten und 55 Sekunden bei zwei Quarantänestationen (b), der Dritte 28 Minuten und 38 Sekunden bei vier Quarantänestationen (c) und der Letzte 6 Minuten und 46 Sekunden bei acht Quarantänestationen (d).

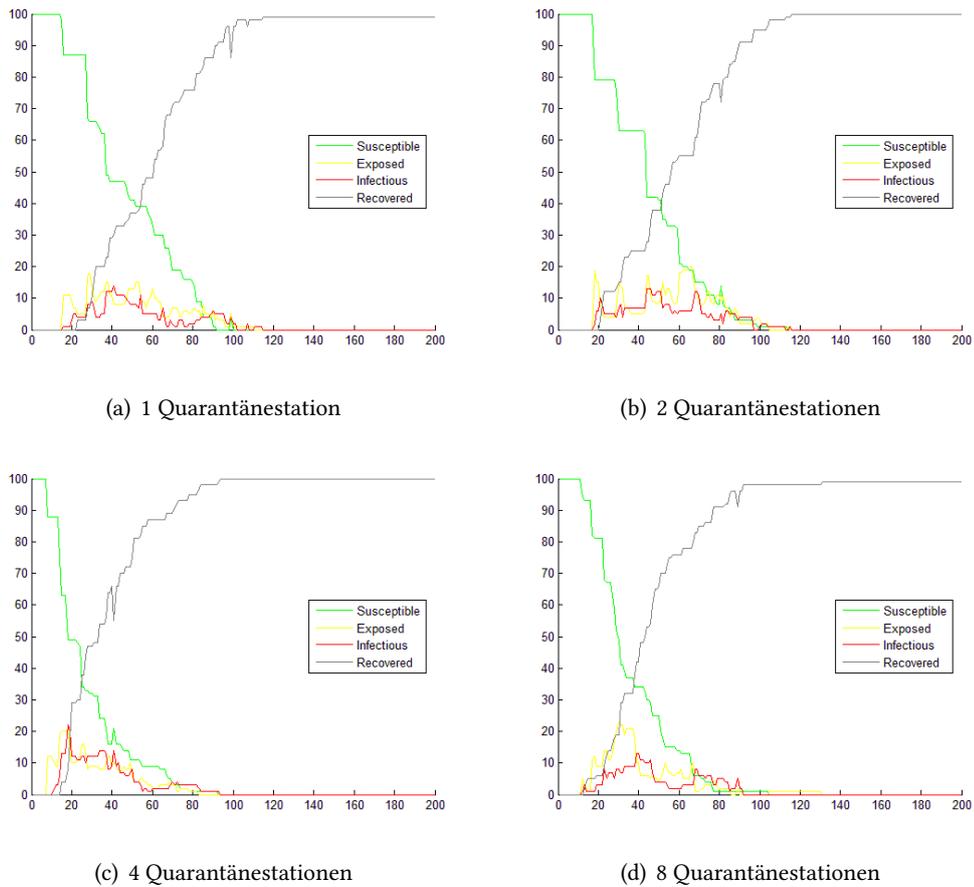


Abbildung 7.11: Unterschiedliche Anzahl an Quarantänestationen - West-Nile-Fieber

Der Epidemie Peak bei einer Quarantänestation (a) ist am Tag 41. An diesem Tag sind 14% der Human Population infiziert. Bei zwei Quarantänestationen (b) ist der Epidemie Peak zwischen den Tagen 44-46. In diesem Zeitraum sind etwa 13% der Human Population infiziert. Am Tag 17 ist der Epidemie Peak bei vier Quarantänestationen (c), an diesen Tag sind etwa 22% der Human Population infiziert. Bei acht Quarantänestationen (d) ist der Epidemie Peak an den Tagen 39-40 und in diesem Zeitraum sind etwa 13% der Human Population infiziert.

### Chikungunya-Fieber

Die Abbildungen 7.12 (a-d) zeigen das Ergebnis von vier Simulationsdurchläufen der tropischen Infektionskrankheit Chikungunya-Fieber. Der erste Simulationsdurchlauf dauerte 36 Minuten und 3 Sekunden bei einer Quarantänestation (a), der Zweite 17 Minuten und 44 Sekunden bei zwei Quarantänestationen (b), der Dritte 56 Minuten und 1 Sekunde bei vier Quarantänestationen (c) und der Letzte 39 Minuten bei acht Quarantänestationen (d).

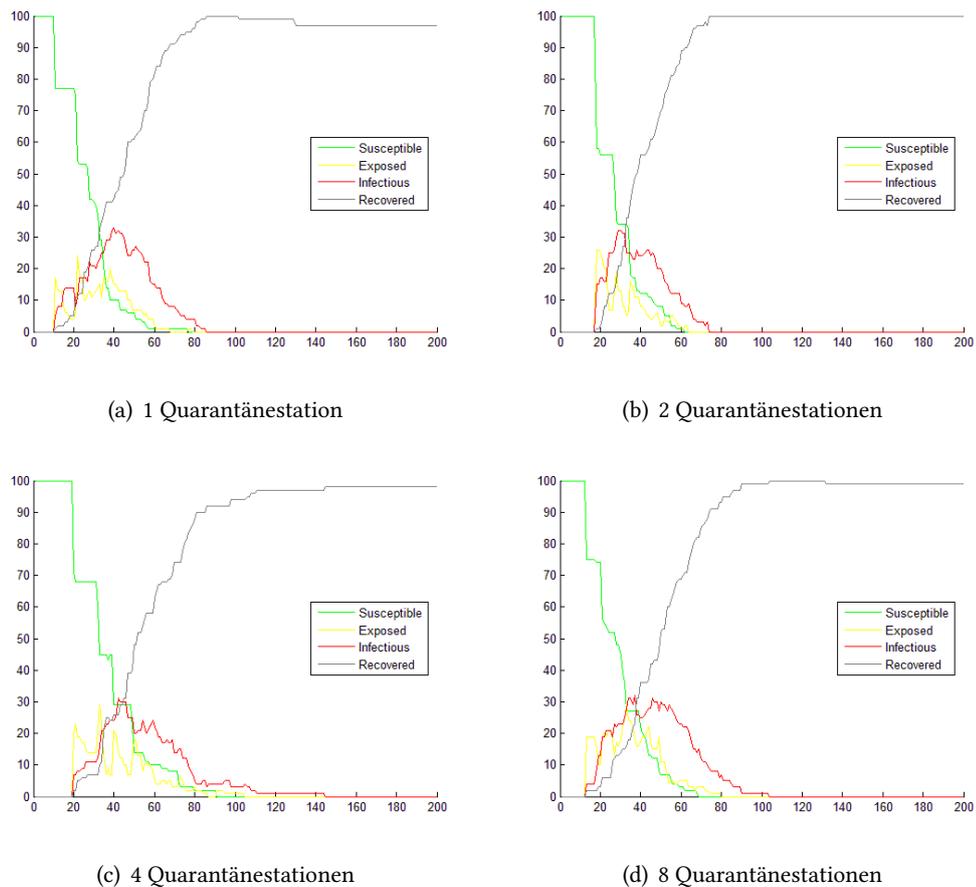


Abbildung 7.12: Unterschiedliche Anzahl an Quarantänestationen - Chikungunya-Fieber

Der Epidemie Peak bei einer Quarantänestation (a) ist am Tag 40. An diesem Tag sind 33% der Human Population infiziert. Bei zwei Quarantänestationen (b) ist der Epidemie Peak zwischen den Tagen 29-31. In diesem Zeitraum sind etwa 32% der Human Population infiziert. Am Tag 43 ist der Epidemie Peak bei vier Quarantänestationen (c), an diesen Tag sind etwa 31% der Human Population infiziert. Bei acht Quarantänestationen (d) ist der Epidemie Peak am Tag 37 und an diesem Tag sind etwa 32% der Human Population infiziert.

### 7.3.3 Schädlingbekämpfer

Bei dem Szenario Schädlingbekämpfer werden 10, 20, 40 und 80 Schädlingbekämpfer definiert. Dies ist im Konfigurationsausschnitt 7.13 zu sehen.

```
1 <CountExterminator >{x}</CountExterminator >
```

Abbildung 7.13: Definition - Count Exterminator

### Dengue-Fieber

Die Abbildungen 7.14 (a-d) zeigen das Ergebnis von vier Simulationsdurchläufen der tropischen Infektionskrankheit Dengue-Fieber. Der erste Simulationsdurchlauf dauerte 50 Minuten und 21 Sekunden bei 10 Schädlingbekämpfern (a), der Zweite 53 Minuten und 56 Sekunden bei 20 Schädlingbekämpfern (b), der Dritte 19 Minuten und 26 Sekunden bei 40 Schädlingbekämpfern (c) und der Letzte 10 Minuten und 36 Sekunden bei 80 Schädlingbekämpfern (d).

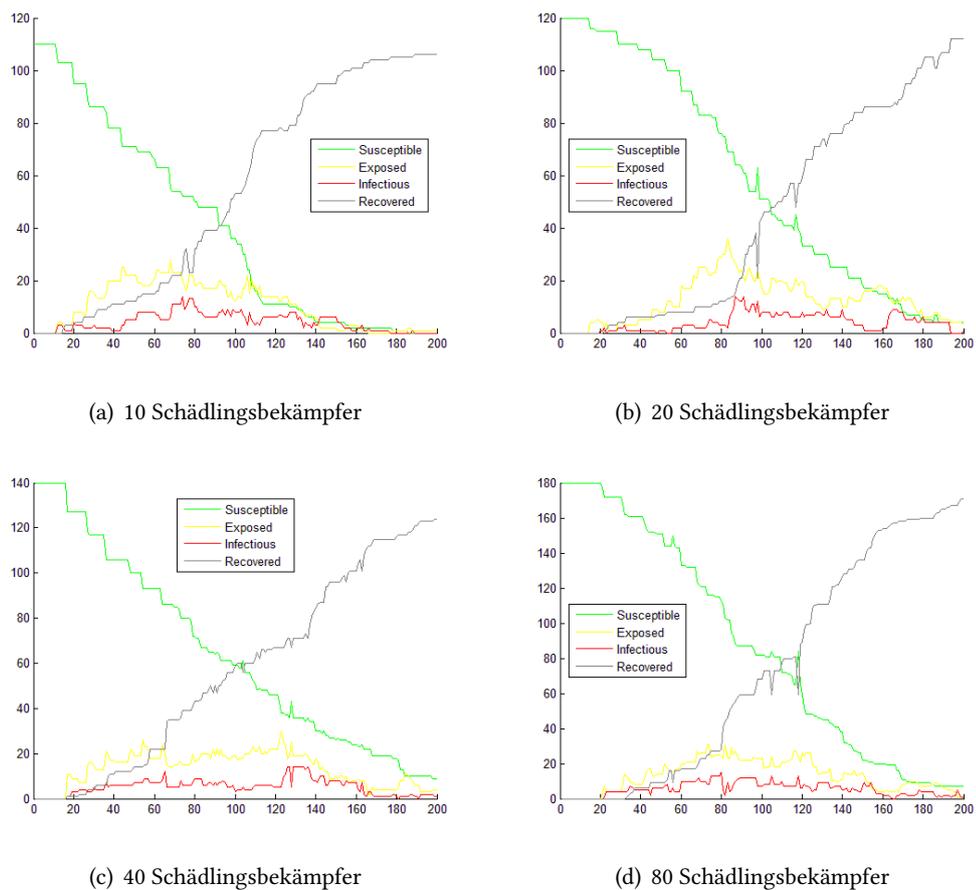
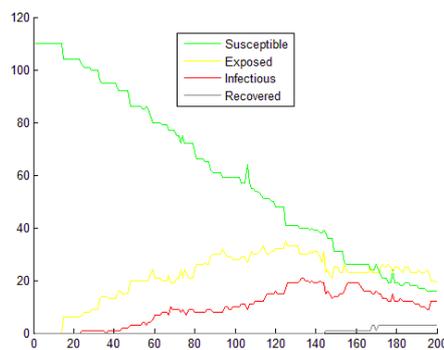


Abbildung 7.14: Unterschiedliche Anzahl an Schädlingbekämpfern - Dengue-Fieber

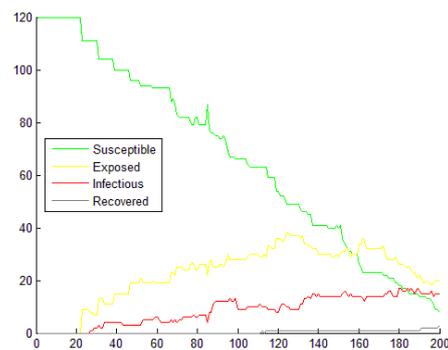
Der Epidemie Peak bei 10 Schädlingbekämpfern (a) ist am Tag 74. An diesem Tag sind 12% der Human Population infiziert. Bei 20 Schädlingbekämpfern (b) ist der Epidemie Peak am Tag 86. An diesem Tag sind etwa 11% der Human Population infiziert. An den Tagen 125-126 ist der Epidemie Peak mit 40 Schädlingbekämpfern (c), an diesen Tagen sind etwa 10% der Human Population infiziert. Bei 80 Schädlingbekämpfern (d) ist der Epidemie Peak am Tag 80 und an diesem Tag sind etwa 8% der Human Population infiziert.

### Malaria

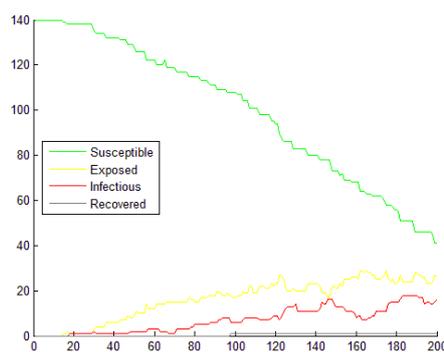
Die Abbildungen 7.15 (a-d) zeigen das Ergebnis von vier Simulationsdurchläufen der tropischen Infektionskrankheit Malaria. Der erste Simulationsdurchlauf dauerte 39 Minuten und 29 Sekunden bei 10 Schädlingbekämpfern (a), der Zweite 11 Minuten und 14 Sekunden bei 20 Schädlingbekämpfern (b), der Dritte 25 Minuten und 13 Sekunden bei 40 Schädlingbekämpfern (c) und der Letzte 21 Minuten und 57 Sekunden bei 80 Schädlingbekämpfern (d).



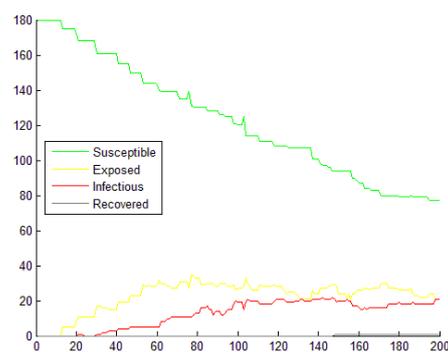
(a) 10 Schädlingbekämpfer



(b) 20 Schädlingbekämpfer



(c) 40 Schädlingbekämpfer



(d) 80 Schädlingbekämpfer

Abbildung 7.15: Unterschiedliche Anzahl an Schädlingbekämpfern - Malaria

Der Epidemie Peak bei 10 Schädlingbekämpfern (a) ist an den Tagen 132-133. An diesem Tag sind 18% der Human Population infiziert. Bei 20 Schädlingbekämpfern (b) ist der Epidemie Peak zwischen den Tagen 180-181. In diesem Zeitraum sind etwa 14% der Human Population infiziert. An den Tagen 184-190 ist der Epidemie Peak mit 40 Schädlingbekämpfern (c), an diesen Tagen sind etwa 12% der Human Population infiziert. Bei 80 Schädlingbekämpfern (d) ist der Epidemie Peak am Tag 143 und an diesem Tag sind etwa 11% der Human Population infiziert.

### West-Nile-Fieber

Die Abbildungen 7.16 (a-d) zeigen das Ergebnis von vier Simulationsdurchläufen der tropischen Infektionskrankheit West-Nile-Fieber. Der erste Simulationsdurchlauf dauerte 56 Minuten und 12 Sekunden bei 10 Schädlingbekämpfern (a), der Zweite 47 Minuten und 9 Sekunden bei 20 Schädlingbekämpfern (b), der Dritte 24 Minuten und 39 Sekunden bei 40 Schädlingbekämpfern (c) und der Letzte 20 Minuten und 25 Sekunden bei 80 Schädlingbekämpfern (d).

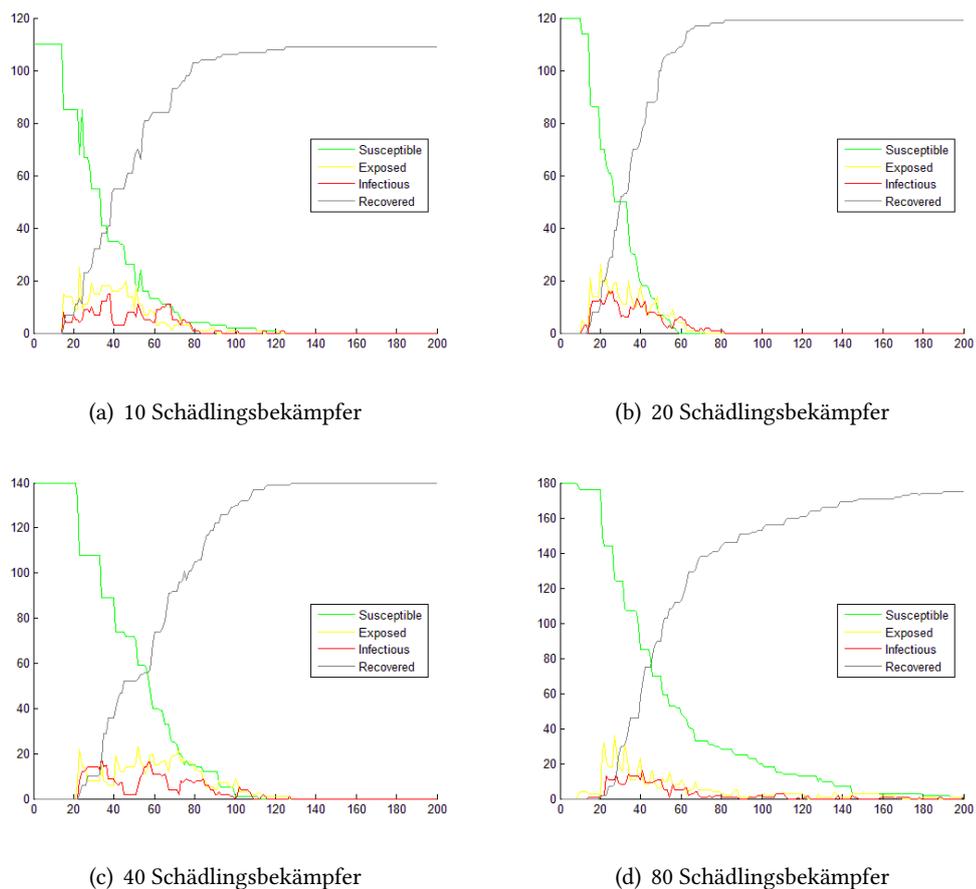


Abbildung 7.16: Unterschiedliche Anzahl an Schädlingbekämpfern - West-Nile-Fieber

Der Epidemie Peak bei 10 Schädlingsbekämpfern (a) ist an den Tagen 37-38. An diesem Tag sind 13% der Human Population infiziert. Bei 20 Schädlingsbekämpfern (b) ist der Epidemie Peak am Tag 24. In diesem Zeitraum sind etwa 13% der Human Population infiziert. Am Tag 34 ist der Epidemie Peak mit 40 Schädlingsbekämpfern (c), an diesen Tag sind etwa 12% der Human Population infiziert. Bei 80 Schädlingsbekämpfern (d) ist der Epidemie Peak am Tag 41 und an diesem Tag sind etwa 9% der Human Population infiziert.

### Chikungunya-Fieber

Die Abbildungen 7.17 (a-d) zeigen das Ergebnis von vier Simulationsdurchläufen der tropischen Infektionskrankheit Chikungunya-Fieber. Der erste Simulationsdurchlauf dauerte 33 Minuten und 28 Sekunden bei 10 Schädlingsbekämpfern (a), der Zweite 14 Minuten und 4 Sekunden bei 20 Schädlingsbekämpfern (b), der Dritte 10 Minuten und 38 Sekunden bei 40 Schädlingsbekämpfern (c) und der Letzte 23 Minuten und 22 Sekunden bei 80 Schädlingsbekämpfern (d).

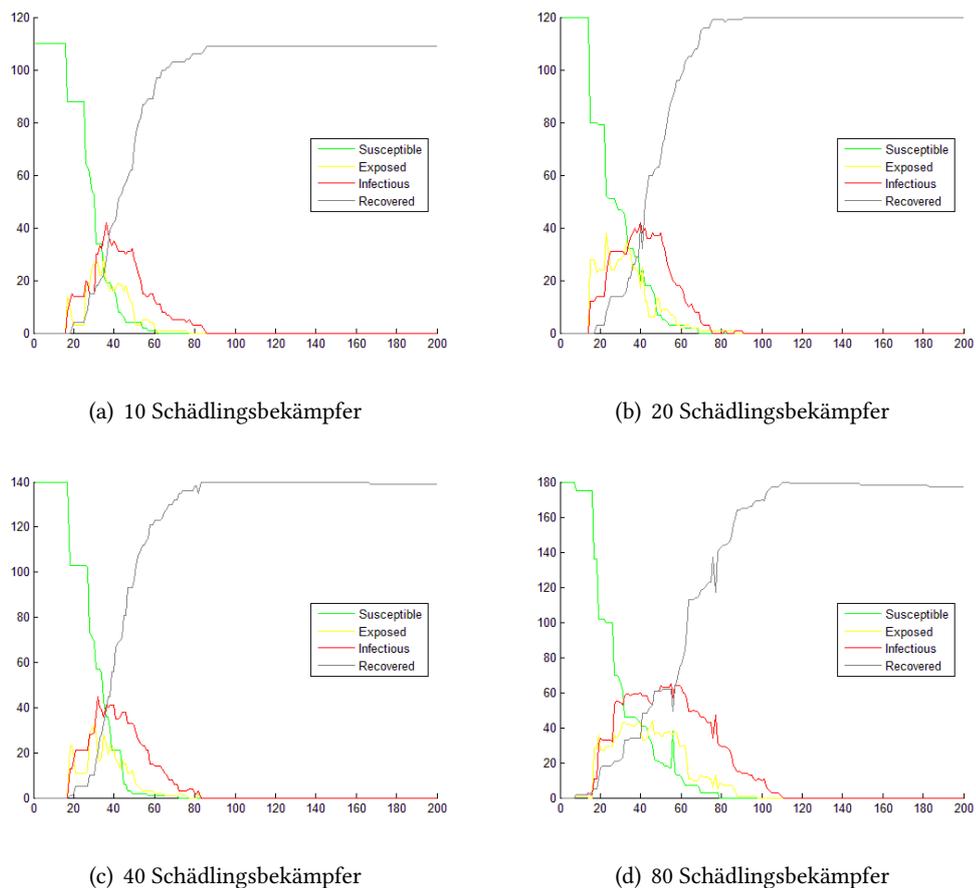


Abbildung 7.17: Unterschiedliche Anzahl an Schädlingsbekämpfern - Chikungunya-Fieber

Der Epidemie Peak bei 10 Schädlingbekämpfern (a) ist am Tag 36. An diesem Tag sind 38% der Human Population infiziert. Bei 20 Schädlingbekämpfern (b) ist der Epidemie Peak am Tag 40. In diesem Zeitraum sind etwa 35% der Human Population infiziert. Am Tag 32 ist der Epidemie Peak mit 40 Schädlingbekämpfern (c), an diesen Tag sind etwa 31% der Human Population infiziert. Bei 80 Schädlingbekämpfern (d) ist der Epidemie Peak am Tag 54 und an diesem Tag sind etwa 35% der Human Population infiziert.

### 7.3.4 Wiederinfizierung

Bei dem Szenario Wiederinfizierung wird der ReInfection-Wert auf *true* bei Wiederinfizierung und auf *false* ohne Wiederinfizierung, gesetzt. Dies ist im Konfigurationsausschnitt 7.18 zu sehen.

```
1 <ReInfection >{ true / false } </ ReInfection >
```

Abbildung 7.18: Definition - ReInfection

### Dengue-Fieber

Die Abbildungen 7.19 (a-b) zeigen das Ergebnis von zwei Simulationsdurchläufen der tropischen Infektionskrankheit Dengue-Fieber. Der erste Simulationsdurchlauf dauerte 44 Minuten und 29 Sekunden mit Wiederinfizierung (a), der Zweite 50 Minuten und 12 Sekunden ohne Wiederinfizierung (b).

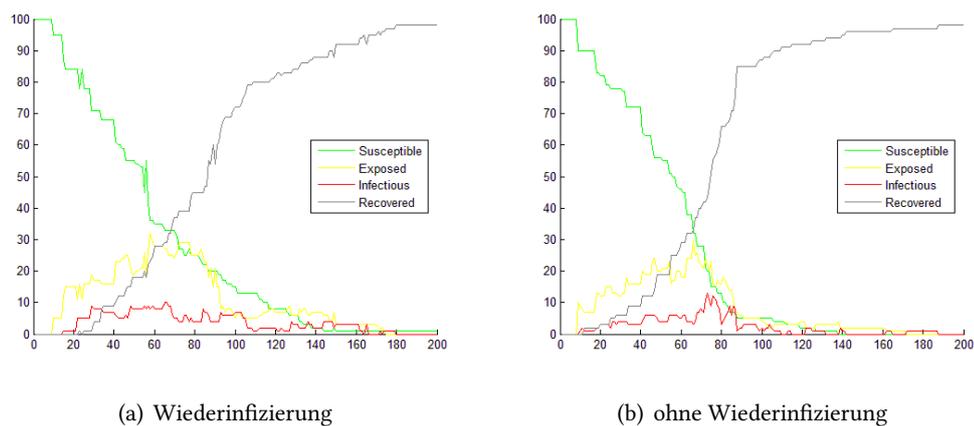


Abbildung 7.19: Wiederinfizierung und keine Wiederinfizierung - Dengue-Fieber

Der Epidemie Peak bei Wiederinfizierung ist an den Tagen 64-65. An diesen Tagen sind 10% der Human Population infiziert. Ohne Wiederinfizierung ist der Epidemie Peak am Tag 72 und an diesem Tag sind etwa 12% der Human Population infiziert.

### Malaria

Die Abbildungen 7.20 (a-b) zeigen das Ergebnis von zwei Simulationsdurchläufen der tropischen Infektionskrankheit Malaria. Der erste Simulationsdurchlauf (a) dauerte 31 Minuten und 54 Sekunden mit Wiederinfizierung, der Zweite (b) 54 Minuten und 48 Sekunden ohne Wiederinfizierung.

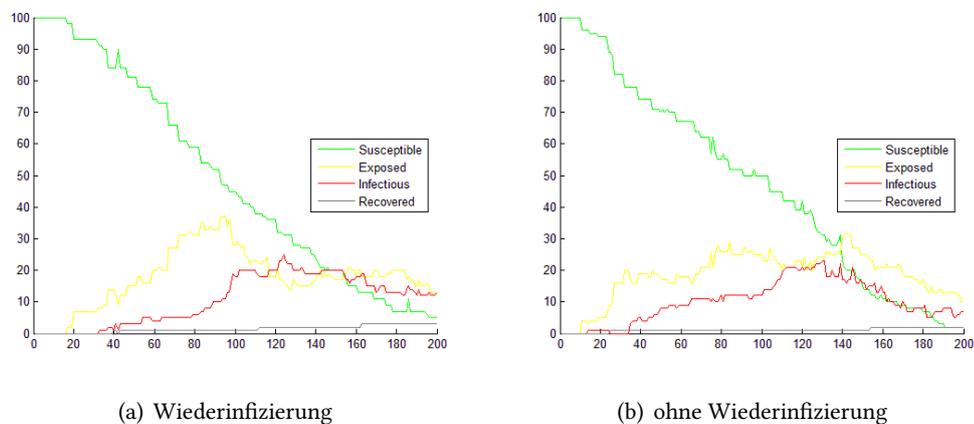


Abbildung 7.20: Wiederinfizierung und keine Wiederinfizierung - Malaria

Der Epidemie Peak bei Wiederinfizierung ist am Tag 124. An diesem Tag sind 25% der Human Population infiziert. Ohne Wiederinfizierung ist der Epidemie Peak am Tag 130 und an diesem Tag sind etwa 22% der Human Population infiziert.

### West-Nile-Fieber

Die Abbildungen 7.21 (a-b) zeigen das Ergebnis von zwei Simulationsdurchläufen der tropischen Infektionskrankheit West-Nile-Fieber. Der erste Simulationsdurchlauf (a) dauerte 59 Minuten und 38 Sekunden mit Wiederinfizierung, der Zweite (b) 10 Minuten und 38 Sekunden ohne Wiederinfizierung.

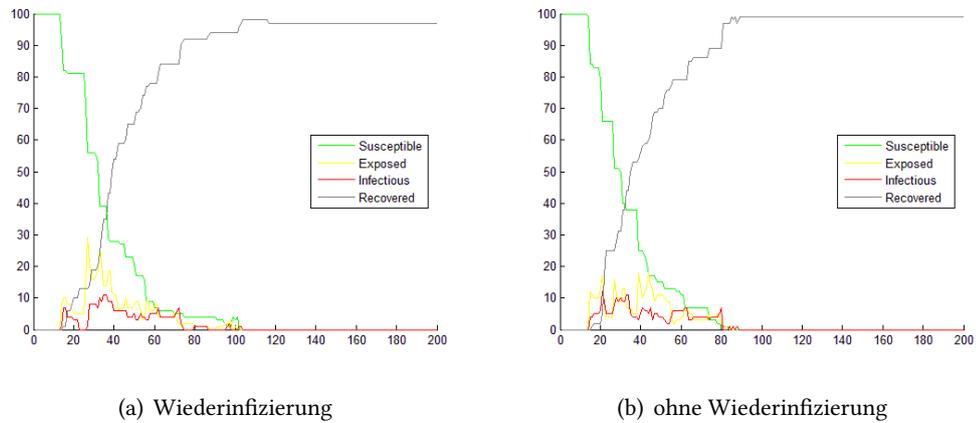


Abbildung 7.21: Wiederinfizierung und keine Wiederinfizierung - West-Nile-Fieber

Der Epidemie Peak bei Wiederinfizierung ist an den Tagen 35-36. An diesen Tagen sind 11% der Human Population infiziert. Ohne Wiederinfizierung ist der Epidemie Peak am Tag 21 und an diesem Tag sind etwa 12% der Human Population infiziert.

### Chikungunya-Fieber

Die Abbildungen 7.22 (a-b) zeigen das Ergebnis von zwei Simulationsdurchläufen der tropischen Infektionskrankheit Chikungunya-Fieber. Der erste Simulationsdurchlauf (a) dauerte 47 Minuten und 31 Sekunden mit Wiederinfizierung, der Zweite (b) 6 Minuten und 41 Sekunden ohne Wiederinfizierung.

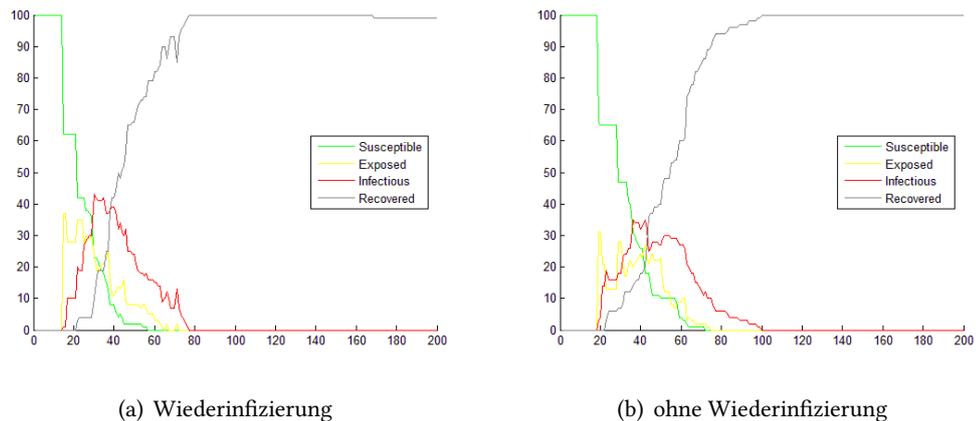


Abbildung 7.22: Wiederinfizierung und keine Wiederinfizierung - Chikungunya-Fieber

Der Epidemie Peak bei Wiederinfizierung ist am Tag 30. An diesem Tag sind 43% der Human Population infiziert. Ohne Wiederinfizierung ist der Epidemie Peak an den Tagen 37-42 und an diesen Tagen sind etwa 34% der Human Population infiziert.

### 7.3.5 Quarantänestationen und Schädlingsbekämpfer

Dieses Szenario besteht aus der Kombination Quarantänestationen und Schädlingsbekämpfern. Je tropischer Infektionskrankheit werden die Parameter des jeweils besten Einzelergebnisses benutzt.

#### Dengue-Fieber

Das beste Ergebnis aus dem Szenario Quarantänestationen sind 2 Quarantänestationen und aus dem Szenario Schädlingsbekämpfer 80 Schädlingsbekämpfer. Die Abbildung 7.23 zeigt das Ergebnis mit 2 Quarantänestationen und 80 Schädlingsbekämpfern der tropischen Infektionskrankheit Dengue-Fieber. Der Simulationsdurchlauf dauerte 49 Minuten und 17 Sekunden.

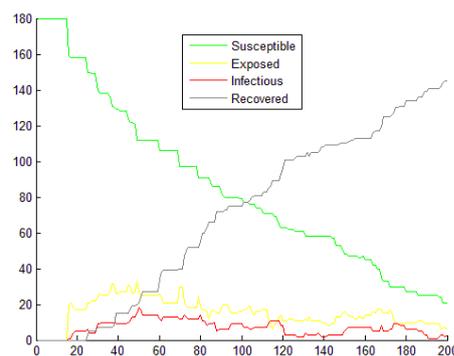


Abbildung 7.23: Dengue-Fieber - 2 Quarantänestationen und 80 Schädlingsbekämpfern

Der Epidemie Peak bei 2 Quarantänestationen und 80 Schädlingsbekämpfern ist am Tag 50. An diesem Tag sind 10% der Human Population infiziert.

#### Malaria

Das beste Ergebnis aus dem Szenario Quarantänestationen sind 4 Quarantänestationen und aus dem Szenario Schädlingsbekämpfer 80 Schädlingsbekämpfer. Die Abbildung 7.24 zeigt das Ergebnis mit 4 Quarantänestationen und 80 Schädlingsbekämpfern der tropischen Infektionskrankheit Malaria. Der Simulationsdurchlauf dauerte 50 Minuten und 11 Sekunden.

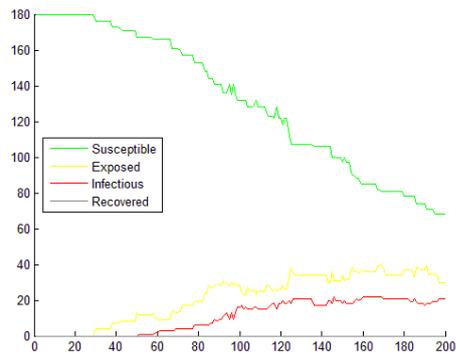
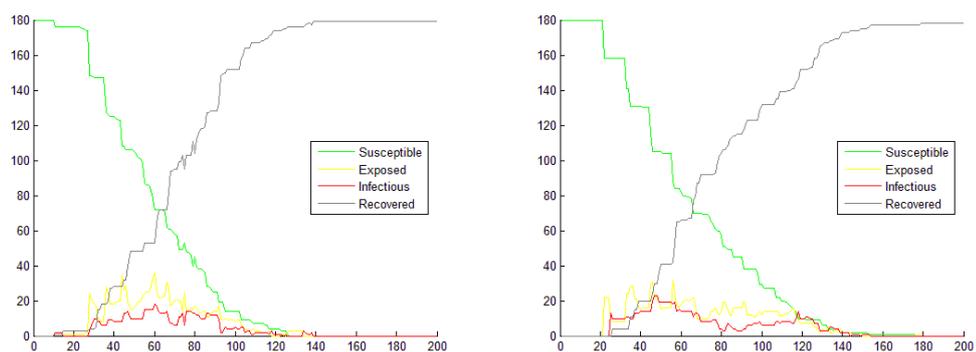


Abbildung 7.24: Malaria - 4 Quarantänestationen und 80 Schädlingsbekämpfern

Der Epidemie Peak bei 4 Quarantänestationen und 80 Schädlingsbekämpfern ist am Tag 145. An diesem Tag sind 12% der Human Population infiziert.

### West-Nile-Fieber

Das beste Ergebnis aus dem Szenario Quarantänestationen sind 2 und 4 Quarantänestationen und aus dem Szenario Schädlingsbekämpfer 80 Schädlingsbekämpfer. Die Abbildung 7.25 zeigt das Ergebnis mit 2 (a) und 4 (b) Quarantänestationen und jeweils 80 Schädlingsbekämpfer der tropischen Infektionskrankheit West-Nile-Fieber. Der erste Simulationsdurchlauf mit 2 Quarantänestationen und 80 Schädlingsbekämpfern (a) dauerte 46 Minuten und 25 Sekunden an und der zweite mit 4 Quarantänestationen und 80 Schädlingsbekämpfern (b) dauerte 21 Minuten und 58 Sekunden.



(a) West-Nile-Fieber - 2 Quarantänestationen und 80 (b) West-Nile-Fieber - 4 Quarantänestationen und 80 Schädlingsbekämpfern

Abbildung 7.25: West-Nile-Fieber - 2/4 Quarantänestationen und 80 Schädlingsbekämpfern

Der Epidemie Peak bei 2 Quarantänestationen und 80 Schädlingsbekämpfern ist am Tag 60. An diesem Tag sind 10% der Human Population infiziert. Bei 4 Quarantänestationen und 80 Schädlingsbekämpfern ist der Epidemie Peak zwischen den Tagen 46-48 und es sind etwa 12% der Human Population infiziert.

### Chikungunya-Fieber

Das beste Ergebnis aus dem Szenario Quarantänestationen sind 4 Quarantänestationen und aus dem Szenario Schädlingsbekämpfer 40 Schädlingsbekämpfer. Die Abbildung 7.26 zeigt das Ergebnis mit 4 Quarantänestationen und 40 Schädlingsbekämpfern der tropischen Infektionskrankheit Chikungunya-Fieber. Der Simulationsdurchlauf dauerte 52 Minuten und 53 Sekunden.

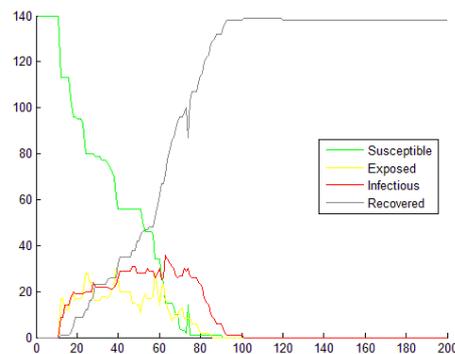


Abbildung 7.26: Chikungunya-Fieber - 4 Quarantänestationen und 40 Schädlingsbekämpfern

Der Epidemie Peak bei 4 Quarantänestationen und 40 Schädlingsbekämpfern ist am Tag 63. An diesem Tag sind 25% der Human Population infiziert.

### Zusammenfassung

Die Tabelle 7.7 (siehe nächste Seiten) zeigt eine Zusammenfassung aller Ergebnisse nach Szenarien und die Tabelle 7.8 (siehe darauffolgende Seiten) eine Zusammenfassung aller Ergebnisse nach Krankheiten aus diesem Abschnitt.

Szenario	Krankheit	Parameter	Simulationszeit	Epidemie Peak/Tag(e)	Infizierte Humans
Default	Dengue-Fieber	-	00:50:12	72	12%
	Malaria	-	00:54:48	130	22%
	West-Nile-Fieber	-	00:10:38	21	12%
	Chikungunya-Fieber	-	00:06:41	37-42	34%
Quarantänestationen	Dengue-Fieber	1	00:40:48	79	14%
	Malaria	1	00:32:50	124	26%
	West-Nile-Fieber	1	00:23:22	41	14%
	Chikungunya-Fieber	1	00:36:03	40	33%
	Dengue-Fieber	2	00:31:22	39-45	11%
	Malaria	2	00:39:58	78-79	23%
	West-Nile-Fieber	2	00:12:55	44-46	13%
	Chikungunya-Fieber	2	00:17:44	29-31	32%
	Dengue-Fieber	4	00:42:36	62	13%
	Malaria	4	00:28:02	173-177	20%
	West-Nile-Fieber	4	00:28:38	17	22%
	Chikungunya-Fieber	4	00:56:01	43	31%
	Dengue-Fieber	8	00:25:02	75	13%
	Malaria	8	00:14:34	160	18%
	West-Nile-Fieber	8	00:06:46	39-40	13%
	Chikungunya-Fieber	8	00:39:00	37	32%

Szenario	Krankheit	Parameter	Simulationszeit	Epidemie Peak/Tag(e)	Infizierte Humans
Schädlingsbekämpfer	Dengue-Fieber	10	00:50:21	74	12%
	Malaria	10	00:39:29	132-133	18%
	West-Nile-Fieber	10	00:56:12	37-38	13%
	Chikungunya-Fieber	10	00:33:28	36	38%
	Dengue-Fieber	20	00:53:56	86	11%
	Malaria	20	00:11:14	180-181	14%
	West-Nile-Fieber	20	00:47:09	24	13%
	Chikungunya-Fieber	20	00:14:04	40	35%
	Dengue-Fieber	40	00:19:26	125-126	10%
	Malaria	40	00:25:13	184-190	12%
	West-Nile-Fieber	40	00:24:39	34	12%
	Chikungunya-Fieber	40	00:10:38	32	31%
	Dengue-Fieber	80	00:10:36	80	8%
	Malaria	80	00:21:57	143	11%
Wiederinfizierung	West-Nile-Fieber	80	00:20:25	41	9%
	Chikungunya-Fieber	80	00:23:22	54	35%
	Dengue-Fieber	Mit	00:44:29	64-65	10%
	Malaria	Mit	00:31:54	124	25%
	West-Nile-Fieber	Mit	00:59:38	35-36	11%
	Chikungunya-Fieber	Mit	00:47:31	30	43%

Szenario	Krankheit	Parameter	Simulationszeit	Epidemie Peak/Tag(e)	Infizierte Humans
	Dengue-Fieber	Ohne	00:50:12	72	12%
	Malaria	Ohne	00:54:48	130	22%
	West-Nile-Fieber	Ohne	00:10:38	21	12%
	Chikungunya-Fieber	Ohne	00:06:41	37-42	34%
Quarantänestationen und Schädlingsbekämpfer	Dengue-Fieber	2 & 80	00:49:17	50	10%
	Malaria	4 & 80	00:50:11	145	12%
	West-Nile-Fieber	2 & 80	00:46:25	60	10%
	West-Nile-Fieber	4 & 80	00:21:58	46-48	12%
	Chikungunya-Fieber	4 & 40	00:52:53	63	25%

Tabelle 7.7: Zusammenfassung der Ergebnisse nach Szenarien

Krankheit	Szenario	Parameter	Simulationszeit	Epidemie Peak/Tag(e)	Infizierte Humans	
Dengue-Fieber	Default	-	00:50:12	72	12%	
	Quarantänestationen	1	00:40:48	79	14%	
	Quarantänestationen	2	00:31:22	39-45	11%	
	Quarantänestationen	4	00:42:36	62	13%	
	Quarantänestationen	8	00:25:02	75	13%	
	Schädlingsbekämpfer	10	00:50:21	74	12%	
	Schädlingsbekämpfer	20	00:53:56	86	11%	
	Schädlingsbekämpfer	40	00:19:26	125-126	10%	
	Schädlingsbekämpfer	80	00:10:36	80	8%	
	Wiederinfizierung	Mit	00:44:29	64-65	10%	
	Wiederinfizierung	Ohne	00:50:12	72	12%	
	Quarantänestationen und Schädlingsbekämpfer	2 & 80	00:49:17	50	10%	
	Malaria	Default	-	00:54:48	130	22%
		Quarantänestationen	1	00:32:50	124	26%
Quarantänestationen		2	00:39:58	78-79	23%	
Quarantänestationen		4	00:28:02	173-177	20%	
Quarantänestationen		8	00:14:34	160	18%	
Schädlingsbekämpfer		10	00:39:29	132-133	18%	
Schädlingsbekämpfer		20	00:11:14	180-181	14%	

Krankheit	Szenario	Parameter	Simulationszeit	Epidemie Peak/Tag(e)	Infizierte Humans
West-Nile-Fieber	Schädlingsbekämpfer	40	00:25:13	184-190	12%
	Schädlingsbekämpfer	80	00:21:57	143	11%
	Wiederinfizierung	Mit	00:31:54	124	25%
	Wiederinfizierung	Ohne	00:54:48	130	22%
	Quarantänestationen und Schädlingsbekämpfer	4 & 80	00:50:11	145	12%
	Default	-	00:10:38	21	12%
	Quarantänestationen	1	00:23:22	41	14%
	Quarantänestationen	2	00:12:55	44-46	13%
	Quarantänestationen	4	00:28:38	17	22%
	Quarantänestationen	8	00:06:46	39-40	13%
	Schädlingsbekämpfer	10	00:56:12	37-38	13%
	Schädlingsbekämpfer	20	00:47:09	24	13%
	Schädlingsbekämpfer	40	00:24:39	34	12%
	Schädlingsbekämpfer	80	00:20:25	41	9%
Wiederinfizierung	Mit	00:59:38	35-36	11%	
Wiederinfizierung	Ohne	00:10:38	21	12%	
Quarantänestationen und Schädlingsbekämpfer	2 & 80	00:46:25	60	10%	

Krankheit	Szenario	Parameter	Simulationszeit	Epidemie Peak/Tag(e)	Infizierte Humans
Chikungunya-Fieber	Quarantänestationen und Schädlingsbekämpfer	4 & 80	00:21:58	46-48	12%
	Default	-	00:06:41	37-42	34%
	Quarantänestationen	1	00:36:03	40	33%
	Quarantänestationen	2	00:17:44	29-31	32%
	Quarantänestationen	4	00:56:01	43	31%
	Quarantänestationen	8	00:39:00	37	32%
	Schädlingsbekämpfer	10	00:33:28	36	38%
	Schädlingsbekämpfer	20	00:14:04	40	35%
	Schädlingsbekämpfer	40	00:10:38	32	31%
	Schädlingsbekämpfer	80	00:23:22	54	35%
	Wiederinfizierung	Mit	00:47:31	30	43%
	Wiederinfizierung	Ohne	00:06:41	37-42	34%
	Quarantänestationen und Schädlingsbekämpfer	4 & 40	00:52:53	63	25%

Tabelle 7.8: Zusammenfassung der Ergebnisse nach Krankheiten

## 7.4 Diskussion

Die nächsten Abschnitte erläutern die Ergebnisse, je tropischer Infektionskrankheit, aus den vorherigen Abschnitten unter **Ergebnisse** (7.3) und eine zusammenfassende Betrachtung der erläuterten Ergebnisse. Jede Simulation wurde nur einmal ausgeführt, deswegen können keine statistischen Abweichungen angegeben und berücksichtigt werden.

### 7.4.1 Dengue-Fieber

Das Szenario Schädlingsbekämpfer reduziert deutlich die Population an infizierten Agenten von 12% auf 8%, das in den Abbildungen 7.14 zu sehen ist. Diese Reduzierung ist darauf zurückzuführen, dass die Schädlingsbekämpfer die Moskitos töten, dadurch sinkt das Risiko gestochen zu werden. Beim dem Szenario Quarantänestationen ist keine ersichtliche Reduzierung der infizierten Agenten zu sehen, der Wert fällt durch die Unterschiede in den einzelnen Simulation unterschiedlich aus (11%, 13% und 14%) (siehe Abbildungen 7.9). In den Quarantänestationen ist die Population vor äußeren Einflüssen geschützt, durch diesen Schutz wird die Krankheit aber nicht behandelt, da es keine Impfstoffe (siehe Abschnitt 2.2.1) gegen die Krankheit gibt. Das Szenario Wiederinfizierung<sup>2</sup> bringt keine ersichtliche Veränderung, der Wert sinkt von 12% (Default) auf 10% infizierte Agenten. Die Kombination aus den Szenarien Quarantänestationen und Schädlingsbekämpfern hat keine deutliche Reduzierung der infizierten Agenten (10%) gebracht (siehe Abbildung 7.23). Zusammenfassend lässt sich sagen, dass ohne Kontrollstrategien (Quarantänestationen und/oder Schädlingsbekämpfern) sich die Population an infizierten Agenten erhöht. Deshalb hat sich die Kontrollstrategie Schädlingsbekämpfer (8%) am Besten bewährt.

Vergleichsweise wurden Werte aus gemeldeten Fällen herangezogen (Vgl. **World Health Organization** (2015d), **World Health Organization** (2015c)). Im Jahre 2010 wurden 1500 Fälle (0,00086% (Vgl. **World Bank Group** (2015)) Infizierte) in Pakistan und im Jahre 2012 209 Fälle (0,00198% (Vgl. **World Bank Group** (2015)) Infizierte) in Portugal gemeldet. Werden diese Werte prozentual dem Wert der Simulation gegenübergestellt fällt auf, dass der simulierte Wert verhältnismäßig hoch ausfällt (12%).

### 7.4.2 Malaria

Das Szenario Schädlingsbekämpfer reduziert deutlich die Population an infizierten Agenten von 18% auf 11%, wie es in den Abbildungen 7.15 zu sehen ist. Diese Reduzierung ist darauf

---

<sup>2</sup>Erneuter Durchlauf des SEIR-Modells nach einem Biss des Moskitos

zurückzuführen, dass die Schädlingsbekämpfer die Moskitos töten, dadurch sinkt das Risiko gestochen zu werden. Beim dem Szenario Quarantänestationen ist auch eine deutliche Reduzierung der infizierten Agenten zu sehen, der Wert sinkt von 26% auf 18% (siehe Abbildungen 7.10), dass ist darauf zurückzuführen, dass die Anzahl der Quarantänestationen steigt und somit mehr Agenten die Chance haben zu überleben. In den Quarantänestationen ist die Population vor äußeren Einflüssen geschützt. Das Szenario Wiederinfizierung bringt auch eine deutliche Veränderung, der Wert sinkt von 22% (Default) auf 10% infizierte Agenten. Die Kombination aus den Szenarien Quarantänestationen und Schädlingsbekämpfer hat eine deutliche Reduzierung der infizierten Agenten (12%) gebracht (siehe Abbildung 7.24). Zusammenfassend lässt sich sagen, dass ohne Kontrollstrategien (Quarantänestationen und/oder Schädlingsbekämpfern) sich die Population an infizierten Agenten erhöht. Deshalb hat sich die Kontrollstrategie Schädlingsbekämpfer (11%) am Besten bewährt, dicht gefolgt von der Kombination aus den Kontrollstrategien Quarantänestationen und Schädlingsbekämpfern (12%).

Vergleichsweise wird ein Wert aus gemeldeten Fällen herangezogen (Vgl. [World Health Organization \(2015e\)](#)). Im Jahre 2006 wurden 280 Fälle (0,0105% (Vgl. [World Bank Group \(2015\)](#)) Infizierte) auf Jamaika gemeldet. Wird dieser Wert prozentual dem Wert der Simulation gegenübergestellt fällt auf, dass der simulierte Wert verhältnismäßig hoch ausfällt (12%).

### 7.4.3 West-Nile-Fieber

Das Szenario Schädlingsbekämpfer reduziert deutlich die Population an infizierten Agenten von 13% auf 9%, das in den Abbildungen 7.16 zu sehen ist. Diese Reduzierung ist darauf zurückzuführen, dass die Schädlingsbekämpfer die Moskitos töten, dadurch sinkt das Risiko gestochen zu werden. Beim dem Szenario Quarantänestationen ist keine ersichtliche Reduzierung der infizierten Agenten zu sehen, der Wert fällt durch die Unterschiede in den einzelnen Simulation unterschiedlich aus (13%, 14% und 22%) (siehe Abbildungen 7.11). In den Quarantänestationen ist die Population vor äußeren Einflüssen geschützt, durch diesen Schutz wird die Krankheit nicht behandelt, da es keine Impfstoffe (siehe Abschnitt 2.2.1) gegen die Krankheit gibt. Das Szenario Wiederinfizierung bringt keine deutliche Veränderung, der Wert sinkt von 12% (Default) auf 11% infizierte Agenten. Die Kombination aus den Szenarien Quarantänestationen und Schädlingsbekämpfern hat nicht das erwünschte Ergebnis einer deutlichen Reduzierung der infizierten Agenten (10%) (siehe Abbildung 7.25) gebracht. Trotz dieser Maßnahmen (Kombination aus Quarantänestationen und Schädlingsbekämpfern) kann nur eine geringe Reduzierung der infizierten Agenten festgestellt werden. Zusammenfassend lässt sich sagen, dass ohne Kontrollstrategien (Quarantänestationen und/oder Schädlingsbekämpfern) sich die Population

an infizierten Agenten erhöht. Deshalb hat sich die Kontrollstrategie Schädlingsbekämpfer (9%) am Besten bewährt.

Vergleichsweise wird ein Wert aus gemeldeten Fällen herangezogen (Vgl. [World Health Organization \(2015f\)](#), [World Health Organization \(2015g\)](#)). Im Jahre 2011 wurden 42 Fälle (Albanien:2, Griechenland:22, Israel:6, Rumänien:1 und russische Föderation:11) (0,0000227% (Vgl. [World Bank Group \(2015\)](#)) Infizierte) in Europa und 2014 1 Fall (0,0000005% (Vgl. [Statista \(2015\)](#)) Infizierte) in Brasilien gemeldet. Werden diese Werte prozentual dem Wert der Simulation gegenübergestellt fällt auf, dass der simulierte Wert verhältnismäßig hoch ausfällt (12%).

### 7.4.4 Chikungunya-Fieber

Das Szenario Schädlingsbekämpfer zeigt keine deutliche Reduzierung an infizierten Agenten, der Wert schwankt zwischen 31% und 38%, das in den Abbildungen [7.17](#) zu sehen ist. Beim dem Szenario Quarantänestationen ist auch keine Reduzierung der infizierten Agenten zu sehen, der Wert fällt durch die Unterschiede in den einzelnen Simulation unterschiedlich aus (31%, 32% und 33%) (siehe Abbildungen [7.12](#)). In den Quarantänestationen ist die Population vor äußeren Einflüssen geschützt, durch diesen Schutz wird die Krankheit nicht behandelt, da es keine Impfstoffe (siehe Abschnitt [2.2.1](#)) gegen die Krankheit gibt. Das Szenario Wiederinfizierung bringt eine Verschlechterung, der Wert steigt von 34% (Default) auf 43% infizierte Agenten. Die Kombination aus den Szenarien Quarantänestationen und Schädlingsbekämpfern hat eine deutliche Reduzierung der infizierten Agenten (25%) gebracht (siehe Abbildung [7.26](#)). Zusammenfassend lässt sich sagen, dass ohne die Kontrollstrategie (Quarantänestationen und/oder Schädlingsbekämpfern) sich die Population an infizierten Agenten erhöht. Deshalb hat sich die Kontrollstrategie Quarantänestationen und Schädlingsbekämpfer (25%) am Besten bewährt.

Vergleichsweise wird ein Wert aus gemeldeten Fällen herangezogen (Vgl. [World Health Organization \(2015a\)](#), [World Health Organization \(2015b\)](#)). Im Jahre 2013 wurden 2 Fälle (0,005% (Vgl. [World Bank Group \(2015\)](#)) Infizierte) auf der St. Martin Insel (Karibik) und 2014 4 Fälle (0,00000628% (Vgl. [Statista \(2015\)](#)) Infizierte) in Frankreich gemeldet. Werden diese Werte prozentual dem Wert der Simulation gegenübergestellt fällt auf, dass der simulierte Wert verhältnismäßig hoch ausfällt (34%).

## Fazit

Viele Faktoren beeinflussen maßgeblich die Simulation, die im Folgendem erläutert werden.

Die Moskitos entfernen sich nicht weit (400 m) von ihrem Geburtsort, d.h. sie bewegen sich lokal um eine Wasserstelle, anstatt landesweit. Daher bilden sich in Gebieten mit Wasserstellen Infektionsherde. Der Biss bzw. die Übertragung des Virus erfolgt, durch einen infizierten weiblichen Moskito, zu bestimmten Tageszeiträumen (Morgens/Abends) (Vgl. [World Health Organization \(2014c\)](#)). Die Schädlingsbekämpfer sind normalerweise nicht nur einmal die Woche aktiv, in der Realität werden sie aktiv, wenn die Konzentration in den betroffenen Gebieten hoch ist.

Außerdem spielt auch das Wetter, im Bezug auf Temperatur und Niederschlag, eine große Rolle und ist ein maßgeblicher Faktor, der das Überleben der Moskitos und deren Nachkommen (Geburten) beeinflusst. Diese Faktoren wurden bei der Simulation nicht berücksichtigt.

Zudem wurden in der Simulation das Alter und Geschlecht der Agenten nicht betrachtet. Ferner können die Abweichungen damit zusammenhängen, dass es eine sehr hohe Dunkelziffer gibt. Diese entsteht, wenn viele Agenten nach einem Biss eines Moskitos nicht zum Arzt gehen und dadurch nicht gelistet werden.

Die Anzahl der Agenten beträgt in der Simulation 10.000, in der Realität beträgt die Anzahl der Agenten in Pakistan 171,73 Mio., Portugal 10,6 Mio., Jamaika 2,7 Mio., Brasilien 200 Mio., Europa 185,76 Mio., St. Martin 31.260 und Frankreich 63,95 Mio. (Vgl. [World Bank Group \(2015\)](#), [Statista \(2015\)](#)). Setzt man die Anzahl der Agenten der Simulation ins Verhältnis zu der Anzahl der Gesamtbevölkerung der einzelnen Ländern, ergeben sich angepasste prozentuale Werte der Simulation. Somit sind 0,0401606% Infizierte in Frankreich, 0,015% Infizierte in Pakistan, 0,4216752% Infizierte in Europa, 0,01% Infizierte in Brasilien, 2,835% Infizierte in Jamaika, 2,0988% Infizierte in Portugal und 14,76878% Infizierte in Pakistan.

Die räumliche Verteilung der Agenten und Moskitos in der Simulation sind nicht gleichzusetzen mit der Realität. In der Simulation wurden die Agenten und Moskitos gleichmäßig über die gesamte definierte Fläche verteilt, in der Realität ist das nicht der Fall.

Die Anzahl der Moskitos betrug in der Simulation gleichbleibend 100.000 anfällige und 10.000 infektiöse Moskitos. Diese Anzahl kann nicht mit der Anzahl an Moskitos in der Realität gleichgesetzt werden.

Diese nicht berücksichtigten Faktoren beeinflussen den prozentualen Wert der Simulation enorm.

Zurzeit werden die einzelnen Zustände des SEIR-Epidemiomodells in der Berechnung der

Reproduktionsnummer nicht berücksichtigt und es wird keine laufende Berechnung während der Simulation durchgeführt, daher verändert sich die Reproduktionsnummer nicht und die Auswirkungen von den Maßnahmen gegen die Epidemie können nicht zeitnah dargestellt werden.

# 8 Schluss

Das letzte Kapitel zeigt eine Zusammenfassung dieser Arbeit und einen Ausblick.

## 8.1 Zusammenfassung

Die Arbeit simulierte die Übertragung von tropischen Infektionskrankheiten zwischen Mensch und Mosquito mit Hilfe eines Multi-Agenten Systems. Die Grundlagen zeigten einen Einblick in vektor-übertragende Infektionskrankheiten, u.a. Dengue-Fieber, Malaria, West-Nile-Fieber und Chikungunya-Fieber, in das SIR-Modell und in die Ausbreitungsmodelle Agentenbasierte Modell und zellulärer Automat. Mit den Verwandten Arbeiten wurde einen Einblick in die Arbeiten von [Roche u. a. \(2008\)](#) mit Multi-Agent-Systems in Epidemiology, [Jacintho u. a. \(2010\)](#) mit Agent-Based-Model for the Spread of the Dengue Fever, [Khalil u. a. \(2010\)](#) mit An agent-based modeling for pandemic influenza in Egypt und [Erickson u. a. \(2010\)](#) mit Vektor-born Epidemie Modell, gegeben. Die Analyse beschrieb die Anforderungen des Prototypen und der Entwurf zeigte die Architektur und die Laufzeitsicht des Systems. Das System wurde im Abschnitt Realisierung beschrieben. Zu guter Letzt wurde mit vordefinierten Parametern, je tropischer Infektionskrankheit, und vordefinierten Szenarien simuliert und die Ergebnisse diskutiert. Die Simulation hat gezeigt, dass es abweichende Werte zur Realität gibt, die unter anderem auf die diskutierten Faktoren zurückzuführen sind.

## 8.2 Ausblick

Der Prototyp bildet nur grob die Realität ab. Es müssten einige Verbesserungen an dem System vorgenommen werden. Die Agenten müssen in die Kategorien Baby, Child, Adult und Senior unterteilt werden, um mit dieser Unterteilung eine Abbildung zur Realität annähernd zu erreichen. Durch diese Unterteilung können weitere Aussagen getroffen werden, wie z.B. wie viele Babys/Kinder durch die tropische Infektionskrankheit erkranken oder versterben. Außerdem fehlt bei Agenten und Moskitos die Einteilung in Geschlechter (männlich und weiblich), um in diesen Bereich auch Aussagen treffen zu können. Die Benutzeroberfläche ist noch nicht ganz ausgereift, z.B. durch Vorgabe von Krankheiten und Regionen (Länder)

kann dem Benutzer die Handhabung mit dem System vereinfacht werden. Wenn Agenten gestochen werden, sollen diese in eine Quarantänestation eingewiesen werden und dort gezielt Maßnahmen gegen die tropische Krankheit bekommen.

Die Berechnung der Reproduktionsnummer sollte kontinuierlich erfolgen und die aktuellen Zustände des SEIR-Epidemiemodells berücksichtigen, um Auswirkungen von getroffenen Maßnahmen zeitnah zu repräsentieren.

# A Realisierung

## A.1 C# Code - InEnvironment() und InVirtualField()

```
1 int n = locationCollection.Count;
2 bool side = false; //true = innerhalb, false = außerhalb
3 for (int i = 0, j = n - 1; i < n; j = i++)
4 {
5     if(
6         ((locationCollection[i].GetY() <= location.GetY()) &&
7          (location.GetY() < locationCollection[j].GetY())) ||
8         ((locationCollection[j].GetY() <= location.GetY()) &&
9          (location.GetY() < locationCollection[i].GetY())) &&
10        (location.GetX() <
11         (locationCollection[j].GetX() -
12          locationCollection[i].GetX()) *
13         (location.GetY() - locationCollection[i].GetY()) /
14         (locationCollection[j].GetY() -
15          locationCollection[i].GetY()) +
16         locationCollection[i].GetX())
17    )
18        side = !side;
19 }
20 return side;
```

Listing A.1: C# Code - InEnvironment() und InVirtualField()

## A.2 XML-Struktur für XML-Datei

```
1 <?xml version="1.0"?>
2 <Configuration xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4     <MoskitoDeathRate >0.05 </MoskitoDeathRate >
5     <MoskitoLifeSpan >30 </MoskitoLifeSpan >
6     <MoskitoChildrenSize >1 </MoskitoChildrenSize >
7     <MoskitoTransmissionRate >0.11 </MoskitoTransmissionRate >
8     <MoskitoPopulationS >1000 </MoskitoPopulationS >
9     <MoskitoPopulationI >1000 </MoskitoPopulationI >
10    <HumanDeathRate >3.57E-05 </HumanDeathRate >
11    <HumanDeathRateDisease >0.003 </HumanDeathRateDisease >
12    <HumanGrowthRate >5.8E-05 </HumanGrowthRate >
```

```
13 <HumanLifeSpan >17800 </HumanLifeSpan >
14 <HumanChildrenSize >1 </HumanChildrenSize >
15 <HumanInfectionRate >0.25 </HumanInfectionRate >
16 <HumanTransmissionRate >0.1 </HumanTransmissionRate >
17 <HumanPopulationS >100000 </HumanPopulationS >
18 <ContactRate >0.12 </ContactRate >
19 <CoordinatenListHospital >
20   <CoordinateHospital >
21     <X>26.884268391728746 </X>
22     <Y>27.103583263392842 </Y>
23   </CoordinateHospital >
24   ...
25 </CoordinatenListHospital >
26 <CoordinatenListEnvironment >
27   <CoordinateEnvironment >
28     <X>31.298657650863703 </X>
29     <Y>24.950262950892842 </Y>
30   </CoordinateEnvironment >
31   ...
32 </CoordinatenListEnvironment >
33 </Configuration >
```

Listing A.2: XML-Struktur der XML-Datei

# B Ergebnisse und Diskussion

## B.1 Graph erstellen

```
1 y = [VarName1, VarName2, VarName3, VarName4];
2
3 hold on;
4 plot(y(:,1), 'green');
5 plot(y(:,2), 'yellow');
6 plot(y(:,3), 'red');
7 plot(y(:,4), 'grey');
8 legend('Susceptible', 'Exposed', 'Infectious', 'Recovered', 'Location', 'Best');
9 hold off;
```

Listing B.3: Matlab Code - Grapherstellung

## C Inhalt der beigefügten CD

Auf der beiliegenden CD-Rom befinden sich folgende Daten:

**Ausarbeitung** Diese Ausarbeitung im PDF-Format.

**Code** Der im Rahmen dieser Arbeit erstellte Quellcode:

**Assemblies** Die für den Prototyp benötigten Bibliotheken.

**Prototyp** Der Prototyp mit grafischer Ausgabe und Konsolenausgabe, zur Ausführung auf einem PC.

# Literaturverzeichnis

- [Ammon, A 2012] AMMON, A ; SUERBAUM, Sebastian (Hrsg.) ; HAHN, Helmut (Hrsg.) ; BURCHARD, Gerd-Dieter (Hrsg.) ; KAUFMANN, Stefan H. E. (Hrsg.) ; SCHULZ, Thomas F. (Hrsg.): *Medizinische Mikrobiologie und Infektiologie*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012 (Springer-Lehrbuch). – 151–156 S. – URL <http://www.springerlink.com/index/10.1007/978-3-642-24167-3>. – 20.11.2012. – ISBN 978-3-642-24166-6
- [Beier 2013a] BEIER, K.: *Epidemiologie*. unveröffentlicht. 2013. – 15.02.2013
- [Beier 2013b] BEIER, K.: *Multiagentensysteme in der Epidemiologie*. unveröffentlicht. 2013. – 31.08.2013
- [Beier 2013c] BEIER, K.: *Multiagentensysteme in der Epidemiologie*. unveröffentlicht. 2013. – 28.07.2013
- [Beier 2014a] BEIER, K.: *Ausbreitung von tropischen Infektionskrankheiten*. unveröffentlicht. 2014. – 08.10.2014
- [Beier 2014b] BEIER, K.: *Simulation der Ausbreitung von Denguefieber in tropischen/subtropischen Gebieten*. unveröffentlicht. 2014. – 16.02.2014
- [Erickson u. a. 2010] ERICKSON, Richard A. ; PRESLEY, Steven M. ; ALLEN, Linda J. ; LONG, Kevin R. ; COX, Stephen B.: A dengue model with a dynamic *Aedes albopictus* vector population, URL <http://www.sciencedirect.com/science/article/pii/S0304380010004497>, 2010, S. 2899 – 2908. – 22.04.2014. – ISSN 0304-3800
- [ETH Zürich 2015] ETH ZÜRICH: *Die Mathematik der Ebola-Epidemie*. 2015. – URL <https://www.ethz.ch/de/news-und-veranstaltungen/eth-news/news/2014/10/die-mathematik-der-ebola-epidemie.html>. – 16.03.2015
- [Guardian 2014] GUARDIAN: *Caribbean chikungunya deaths hit 113*. 2014. – URL <http://www.guardian.co.tt/news/2014-09-18/caribbean-chikungunya-deaths-hit-113>. – 19.10.2014

- [Jacintho u. a. 2010] JACINTHO, Luís F. O. ; BATISTA, André F. M. ; RUAS, Terry L. ; MARIETTO, Maria G. B. ; SILVA, Fábio A.: An agent-based model for the spread of the Dengue fever: a swarm platform simulation approach. In: *Proceedings of the 2010 Spring Simulation Multiconference*. San Diego, CA, USA : Society for Computer Simulation International, 2010 (SpringSim '10), S. 1–8. – URL <http://dx.doi.org/10.1145/1878537.1878540>. – 23.04.2013. – ISBN 978-1-4503-0069-8
- [JetBrains 2014] JETBRAINS: *Resharper*. 2014. – URL <https://www.jetbrains.com/resharper/>. – 03.12.2014
- [Khalil u. a. 2010] KHALIL, K.M. ; ABDEL-AZIZ, M. ; NAZMY, T.T. ; SALEM, A.-B.M.: An agent-based modeling for pandemic influenza in Egypt. In: *The 7th International Conference on Informatics and Systems (INFOS), 2010*, URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5461812>, 2010, S. 1 –7. – 04.11.2012
- [Kim Beier 2012] KIM BEIER: *Realisierung eines Software-Leitstandes am Beispiel des HAW-Logistics-Systems*. 2012. – URL [http://edoc.sub.uni-hamburg.de/haw/volltexte/2012/1748/pdf/Bachelorarbeit\\_KimBeier.pdf](http://edoc.sub.uni-hamburg.de/haw/volltexte/2012/1748/pdf/Bachelorarbeit_KimBeier.pdf). – 13.06.2015
- [Macal und North 2010] MACAL, C M. ; NORTH, M J.: Tutorial on agent-based modeling and simulation, Palgrave Macmillan, September 2010, S. 151–162. – URL <http://www.palgrave-journals.com/doi/10.1057/jos.2010.3>. – 11.11.2012. – ISSN 1747-7778
- [Macal und North 2011] MACAL, C.M. ; NORTH, M.J.: Introductory tutorial: Agent-based modeling and simulation. In: *Simulation Conference (WSC), Proceedings of the 2011 Winter*, URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6147864>, 2011, S. 1451 –1464. – 12.12.2012. – ISSN 0891-7736
- [MedicineNet 2014] MEDICINET: *West Nile Virus*. 2014. – URL [http://www.medicinenet.com/west\\_nile\\_encephalitis/article.htm#what\\_is\\_the\\_history\\_of\\_west\\_nile\\_virus](http://www.medicinenet.com/west_nile_encephalitis/article.htm#what_is_the_history_of_west_nile_virus). – 19.10.2014
- [Microsoft 2014] MICROSOFT: *Bing Maps Windows Presentation Foundation (WPF) Control, Version 1.0*. 2014. – URL <http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=27165>. – 03.12.2014
- [Orwant u. a. 2000] ORWANT, J. ; HIETANIEMI, J. ; MACDONALD, J.: *Algorithmen mit Perl*. Kap. 10 Geometrische Algorithmen, O'Reilly, 2000. – URL <http://www>.

- [oreilly.de/catalog/maperlger/chapter/kap10.pdf](http://oreilly.de/catalog/maperlger/chapter/kap10.pdf). – 04.05.2014. – ISBN 9783897211414
- [Paleshi u. a. 2011] PALESHI, A. ; EVANS, G.W. ; HERAGU, S.S. ; MOGHADDAM, K.S.: Simulation of mitigation strategies for a pandemic influenza. In: *Simulation Conference (WSC), Proceedings of the 2011 Winter*, 2011, S. 1340 –1348. – 12.12.2012. – ISSN 0891-7736
- [Qin u. a. 2012] QIN, Hong ; SHAPIRO, Alexander ; YANG, Li: Emerging Infectious Disease: A Computational Multi-agent Model. In: *ASE/IEEE International Conference on Bio-Medical Computing (BioMedCom)*, URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6516424>, 2012, S. 28–33. – 02.06.2013
- [Robert Koch-Institut 2012] ROBERT KOCH-INSTITUT: *Malaria*. 2012. – URL [http://www.rki.de/DE/Content/Infekt/EpidBull/Merkblaetter/Ratgeber\\_Malaria.html#doc2392924bodyText1](http://www.rki.de/DE/Content/Infekt/EpidBull/Merkblaetter/Ratgeber_Malaria.html#doc2392924bodyText1). – 11.10.2014
- [Roche u. a. 2008] ROCHE, Benjamin ; GUÉGAN, Jean-François ; BOUSQUET, François: Multi-agent systems in epidemiology: a first step for computational biology in the study of vector-borne disease transmission. In: *BioMed Central*, URL <http://dx.doi.org/10.1186/1471-2105-9-435>, 2008, S. 1–9. – 30.04.2013
- [Statista 2015] STATISTA: *Länder*. 2015. – URL <http://de.statista.com/statistik/kategorien/kategorie/19/branche/laender/>. – 01.02.2015
- [Sun u. a. 2012] SUN, Jun ; IBRAHIM, Abdoul N. ; GONG, Jianhua ; YANG, Liyang ; LI, Yi ; ZHOU, Jieping: Study on spread of vibrio cholera in rivers based on Cellular Automata Model. In: *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=06351379>, 2012, S. 491 –494. – 12.12.2012. – ISSN 2153-6996
- [World Bank Group 2015] WORLD BANK GROUP: *World DataBank - total Population*. 2015. – URL <http://databank.worldbank.org/data/views/reports/tableview.aspx?isshared=true>. – 01.02.2015
- [World Health Organization 2009] WORLD HEALTH ORGANIZATION: *Frequently-Asked-Questions about Malaria*. 2009. – URL [http://www.afro.who.int/fr/downloads/doc\\_download/5023-frequently-asked-questions-about-malaria.html](http://www.afro.who.int/fr/downloads/doc_download/5023-frequently-asked-questions-about-malaria.html). – 19.10.2014

- [World Health Organization 2011] WORLD HEALTH ORGANIZATION: *West Nile virus*. 2011. – URL <http://www.who.int/mediacentre/factsheets/fs354/en/>. – 11.10.2014
- [World Health Organization 2014a] WORLD HEALTH ORGANIZATION: *Chikungunya*. 2014. – URL <http://www.who.int/mediacentre/factsheets/fs327/en/>. – 11.10.2014
- [World Health Organization 2014b] WORLD HEALTH ORGANIZATION: *Chikungunya*. 2014. – URL [http://www.who.int/denguecontrol/arbo-viral/other\\_arboviral\\_chikungunya/en/](http://www.who.int/denguecontrol/arbo-viral/other_arboviral_chikungunya/en/). – 11.10.2014
- [World Health Organization 2014c] WORLD HEALTH ORGANIZATION: *Dengue*. 2014. – URL <http://www.who.int/denguecontrol/en/>. – 19.10.2014
- [World Health Organization 2014d] WORLD HEALTH ORGANIZATION: *Dengue and severe dengue*. 2014. – URL <http://www.who.int/mediacentre/factsheets/fs117/en/>. – 11.10.2014
- [World Health Organization 2014e] WORLD HEALTH ORGANIZATION: *Malaria*. 2014. – URL <http://www.who.int/mediacentre/factsheets/fs094/en/>. – 11.10.2014
- [World Health Organization 2014f] WORLD HEALTH ORGANIZATION: *Vector-borne diseases - Chikungunya*. 2014. – URL <http://www.who.int/mediacentre/factsheets/fs387/en/index3.html>. – 11.10.2014
- [World Health Organization 2014g] WORLD HEALTH ORGANIZATION: *Vector-borne diseases - Dengue*. 2014. – URL <http://www.who.int/mediacentre/factsheets/fs387/en/index2.html>. – 11.10.2014
- [World Health Organization 2014h] WORLD HEALTH ORGANIZATION: *Vector-borne diseases - Malaria*. 2014. – URL <http://www.who.int/mediacentre/factsheets/fs387/en/index1.html>. – 11.10.2014
- [World Health Organization 2015a] WORLD HEALTH ORGANIZATION: *Global Alert and Response (GAR) - Chikungunya - France*. 2015. – URL <http://www.who.int/csr/don/23-october-2014-chikungunya/en/>. – 30.01.2015

- [World Health Organization 2015b] WORLD HEALTH ORGANIZATION: *Global Alert and Response (GAR) - Chikungunya in the French part of the Caribbean isle of Saint Martin*. 2015. – URL [http://www.who.int/csr/don/2013\\_12\\_10a/en/](http://www.who.int/csr/don/2013_12_10a/en/). – 30.01.2015
- [World Health Organization 2015c] WORLD HEALTH ORGANIZATION: *Global Alert and Response (GAR) - Crimean-Congo haemorrhagic fever (CCHF) and Dengue in Pakistan*. 2015. – URL [http://www.who.int/csr/don/2010\\_10\\_25a/en/](http://www.who.int/csr/don/2010_10_25a/en/). – 30.01.2015
- [World Health Organization 2015d] WORLD HEALTH ORGANIZATION: *Global Alert and Response (GAR) - Dengue Fever in Madeira, Portugal*. 2015. – URL [http://www.who.int/csr/don/2012\\_10\\_17/en/](http://www.who.int/csr/don/2012_10_17/en/). – 30.01.2015
- [World Health Organization 2015e] WORLD HEALTH ORGANIZATION: *Global Alert and Response (GAR) - Malaria in Jamaica*. 2015. – URL [http://www.who.int/csr/don/2007\\_02\\_09/en/](http://www.who.int/csr/don/2007_02_09/en/). – 30.01.2015
- [World Health Organization 2015f] WORLD HEALTH ORGANIZATION: *Global Alert and Response (GAR) - West Nile virus - Brazil*. 2015. – URL <http://www.who.int/csr/don/15-december-2014-wnv/en/>. – 30.01.2015
- [World Health Organization 2015g] WORLD HEALTH ORGANIZATION: *Global Alert and Response (GAR) - West Nile Virus Infection (WNV) in Europe*. 2015. – URL [http://www.who.int/csr/don/2011\\_08\\_16/en/](http://www.who.int/csr/don/2011_08_16/en/). – 30.01.2015

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 28. Juli 2015

---

Kim Beier